BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

AUTOMATIC POINT-FEATURE LABEL PLACEMENT

AUTOMATICKÉ UMÍSŤOVÁNÍ POPISU BODOVÝCH DAT

DOCTORAL THESIS
DISERTAČNÍ PRÁCE

AUTHOR **Ing. PETR BOBÁK**
AUTOR PRÁCE

SUPERVISOR **doc. Ing. MARTIN ČADÍK, Ph.D.**
ŠKOLITEL

BRNO 2024

Abstract

Automatic label placement is a crucial aspect of data visualization, essential for enhancing the clarity and readability of visual representations across various domains such as cartography, medical imaging, and emergency response management. Label, in this context, refers to textual or symbolic annotation that identifies or explains specific point feature within a visualization, such as the name of the city on a map, measurement on a medical scan, or position of AED in a schematic map for emergency response dispatchers. The work covered in this dissertation aims to advance the field by developing novel techniques addressing the inherent challenges associated with internal and external label placement in complex visualizations. Internal label placement refers to placing labels close to the point features they describe within the boundaries of the visualization. External label placement, on the other hand, involves placing labels outside the main visualization area, connected to the relevant features by lines.

Our research focuses on three key areas: achieving temporally stable and visually coherent boundary label placements, leveraging machine learning to improve the completeness of internal label placements, and optimizing label positioning by integrating perceptual insights. The dissertation begins with a comprehensive review of existing techniques, identifying significant gaps in handling dynamic environments and maintaining visual coherence. The literature review also highlights that the label placement quality is not entirely and precisely defined, as many cartographic guidelines rely on best practices rather than empirical studies. Building on these insights, we introduce novel optimization methods for boundary label placement in dynamic panoramic visualizations, minimizing label movement and reducing user cognitive load. Experimental results demonstrate the effectiveness of these approaches in maintaining label stability without compromising readability or clarity. In the context of internal label placement, we explore the relevance of deep reinforcement learning and propose a novel method that significantly improves label completeness, particularly in dense and complex scenarios. Furthermore, we introduce a perceptual study that determines user-preferred label positions, challenges conventional placement strategies, and demonstrates the importance of considering user preferences in label placement design. Our supplementary study on users' preferred label density, a topic scarcely explored in existing literature, further confirms that integrating perceptual insights into the label placement process significantly enhances the overall user experience, leading to more intuitive and compelling visualizations.

While the proposed methods offer substantial improvements over existing techniques, we acknowledge several limitations, including the complexity of implementing the boundary label optimization in real-time scenarios and the computational demands of the reinforcement learning approach. Future research directions include the development of mixed label placement models for 3D visualizations, optimization of computational efficiency, and further exploration of user perception to refine label placement techniques.

Abstrakt

Automatické umísťování popisu je klíčovým aspektem vizualizace dat. Popis je zásadní pro zvýšení srozumitelnosti a čitelnosti vizuálních reprezentací napříč různými oblastmi, jako jsou kartografie, lékařské zobrazování a krizový management. Popisek, v plurálu popis, v tomto kontextu odkazuje na textovou nebo symbolickou anotaci, která identifikuje nebo vysvětluje konkrétní bodový prvek v rámci vizualizace, jako je název města na mapě, hodnota měření na lékařském snímku nebo pozice defibrilátoru v situační mapě pro krizový managment integrovaného záchranného systému. Tato disertace si klade za cíl prohloubit tuto oblast vývojem nových metod, které řeší inherentní výzvy spojené s umísťováním interního a externího popisu v komplexních vizualizacích. Interní popis spočívá v umístění textových nebo symbolických anotací poblíž bodových prvků uvnitř hranic vizualizace. Externí popis naopak odkazuje na umístění mimo hlavní vizualizaci, kde jsou bodové prvky propojeny s příslušným popisem pomocí čar.

Náš výzkum se zaměřuje na tři klíčové oblasti: dosažení časově stabilního a vizuálně koherentního umístění popisu na okraji vizualizace, využití strojového učení ke zvýšení úplnosti popisu a optimalizaci umístění popisu s využitím percepčních poznatků. Disertace poskytuje komplexní přehled stávajících přístupů, který identifikuje významné mezery v řešení dynamických vizualizací a udržování vizuální koherence. Literární přehled také ukazuje, že kvalita umístění popisu není zcela a přesně definována a mnohé kartografické doporučení se spoléhají na zažité postupy spíše než na empirické studie s uživateli. Na základě těchto poznatků jsme představili nové optimalizační metody umísťování popisu na okraj dynamických panoramatických vizualizací, které minimalizují pohyb popisu a snižují kognitivní zátěž uživatele. Experimentální výsledky demonstrují signifikantní zlepšení vizuální koherence popisu bez negativního vlivu na jeho čitelnost nebo jednoznačnost. V kontextu interního umísťování popisu jsme prozkoumali význam hlubokého posilovaného učení a na tomto základě jsme představili novou metodu, která výrazně zlepšuje úplnost popisu, zejména v hustých a komplexních vizualizacích. Dále jsme uspořádali percepční studii, která identifikuje uživatelsky preferované pozice popisu kolem význačných bodů, zpochybňuje konvenční doporučení a vyzdvihuje důležitost zohlednění uživatelských preferencí při umísťování popisu. Naše následná studie zabývající se preferovanou hustotou popisu, což je téma v existující literatuře zřídka zkoumané, dále potvrzuje, že integrace percepčních poznatků do procesu umísťování popisu výrazně zlepšuje celkový uživatelský zážitek, což vede k intuitivnějším a přehlednějším vizualizacím.

Ačkoli představené metody poskytují podstatná zlepšení oproti stávajícím technikám, je potřeba zmínit také jejich limitace, jako je komplexní implementace optimalizační metody umísťování popisu v dynamických vizualizacích a výpočetní náročnost umísťování popisu s využitím posilovaného učení. Budoucí směry výzkumu zahrnují vývoj smíšených modelů umísťování popisů pro 3D vizualizace, optimalizaci výpočetní efektivity a další prohlubovaní znalostí o vnímání popisu uživatelem.

Keywords

Data visualization, automatic label placement, external label placement, internal label placement, machine learning in visualization, deep reinforcement learning, perception, visual coherence, dynamic visualizations, user-centered design, label position optimization

Klíčová slova

Vizualizace dat, automatické umísťování popisu, externí umísťování popisu, interní umísťování popisu, strojové učení ve vizualizaci, hluboké posilované učení, percepce, vizuální koherence, dynamické vizualizace, design zaměřený na uživatele, optimalizace polohy popisu

Reference

BOBÁK, Petr. *Automatic Point-feature Label Placement*. Brno, 2024. Doctoral thesis. Brno University of Technology, Faculty of Information Technology. Supervisor doc. Ing. Martin Čadík, Ph.D.

Rozšířený abstrakt

Automatické umísťování popisu je klíčovým aspektem vizualizace dat. Popis je zásadní pro zvýšení srozumitelnosti a čitelnosti vizuálních reprezentací napříč různými oblastmi, jako jsou kartografie, lékařské zobrazování a krizový management. Popisek, v plurálu popis, v tomto kontextu odkazuje na textovou nebo symbolickou anotaci, která identifikuje nebo vysvětluje konkrétní bodový prvek v rámci vizualizace, jako je název města na mapě, hodnota měření na lékařském snímku nebo pozice defibrilátoru v situační mapě pro krizový managment integrovaného záchranného systému. Tato disertace si klade za cíl prohloubit tuto oblast vývojem nových metod, které řeší inherentní výzvy spojené s umísťováním interního a externího popisu v komplexních vizualizacích. Interní popis spočívá v umístění textových nebo symbolických anotací poblíž bodových prvků uvnitř hranic vizualizace. Externí popis naopak odkazuje na umístění mimo hlavní vizualizaci, kde jsou bodové prvky propojeny s příslušným popisem pomocí čar.

Automatické umísťování popisu nachází uplatnění v širokém spektru oblastí, kde přesnost a rychlost rozhodování hraje klíčovou roli. V lékařském zobrazování automatický popis výrazně usnadňuje anotaci anatomických struktur, což je nezbytné pro diagnostiku, plánování léčby a vzdělávací účely. Anotace pomáhají radiologům v diagnostice obrazových dat, zaznamenávání diagnostických zjištění a usnadňují diskuse při schůzkách lékařských týmů [134]. V řízení krizových situací, jako jsou například operace integrovaného záchranného systému, je správné a rychlé rozhodování často otázkou života a smrti. Jasné a přesné informace na mapách, které dispečeři používají, jsou proto nezbytné. Automatizované umísťování popisu může zlepšit rychlost a přesnost poskytovaných informací, čímž se snižuje riziko chyb a zvyšuje efektivita rozhodování v nouzových situacích [58]. V oblasti kartografie a řízení letového provozu jsou popisky nezbytné pro přesnou identifikaci geografických bodů nebo informací o letadlech. V těchto aplikacích je důležité, aby se popis co nejméně překrýval a byl snadno čitelný, neboť jakákoliv nejasnost může vést k vážným chybám v rozhodování [76]. V technických výkresech, které se používají ve strojním nebo stavebním inženýrství a údržbových manuálech, jsou popisky zásadní pro poskytování informací o součástech a jejich funkcích. Přesné a efektivní označování v těchto výkresech je klíčové pro zajištění bezpečnosti a efektivity při montáži a údržbě [101]. Konečně, ve sportovních vizualizacích, například při zobrazování real-time informací o účastnících závodů nebo sportovních utkání, je dynamické umísťování popisu klíčové pro rozšíření diváckého zážitku. Vizuální popis sportovních přenosů zvyšuje pochopení dané události a zapojení diváku do děje.

Náš výzkum se zaměřuje na tři klíčové oblasti: dosažení časově stabilního a vizuálně koherentního umístění popisu na okraji vizualizace, využití strojového učení ke zvýšení úplnosti popisu a optimalizaci umístění popisu s využitím percepčních poznatků. Disertace poskytuje komplexní přehled stávajících přístupů, který identifikuje významné mezery v řešení dynamických vizualizací a udržování vizuální koherence. Literární přehled také ukazuje, že kvalita umístění popisu není zcela a přesně definována a mnohé kartografické doporučení se spoléhají na zažité postupy spíše než na empirické studie s uživateli. Na základě těchto poznatků jsme představili nové optimalizační metody umísťování popisu na okraj dynamických panoramatických vizualizací, které minimalizují pohyb popisu a snižují kognitivní zátěž uživatele. Experimentální výsledky demonstrují signifikantní zlepšení vizuální koherence popisu bez negativního vlivu na jeho čitelnost nebo jednoznačnost. V kontextu interního umísťování popisu jsme prozkoumali význam hlubokého posilovaného učení a na tomto základě jsme představili novou metodu, která výrazně zlepšuje úplnost popisu, zejména v hustých a komplexních vizualizacích. Dále jsme uspořádali percepční studii,

která identifikuje uživatelsky preferované pozice popisu kolem význačných bodů, zpochybňuje konvenční doporučení a vyzdvihuje důležitost zohlednění uživatelských preferencí při umísťování popisu. Naše následná studie zabývající se preferovanou hustotou popisu, což je téma v existující literatuře zřídka zkoumané, dále potvrzuje, že integrace percepčních poznatků do procesu umísťování popisu výrazně zlepšuje celkový uživatelský zážitek, což vede k intuitivnějším a přehlednějším vizualizacím.

Ačkoli představené metody poskytují podstatná zlepšení oproti stávajícím technikám, je potřeba zmínit také jejich limitace, jako je komplexní implementace optimalizační metody umísťování popisu v dynamických vizualizacích a výpočetní náročnost umísťování popisu s využitím posilovaného učení. Budoucí směry výzkumu zahrnují vývoj smíšených modelů umísťování popisů pro 3D vizualizace, optimalizaci výpočetní efektivity a další prohlubovaní znalostí o vnímání popisu uživatelem.

Automatic Point-feature Label Placement

Declaration

I hereby declare that this doctoral thesis was prepared as an original work by the author under the supervision of doc. Ing. Martin Čadík, Ph.D. I have listed all the literary sources, publications, and other sources which were used during the preparation of this thesis.

.....
Petr Bobák
September 11, 2024

Acknowledgements

First and foremost, I want to thank my supervisor, Martin Čadík, for his exceptional guidance, insightful feedback, and persistent support throughout this journey. His encouragement has been instrumental in shaping the direction and outcome of this dissertation. In addition, I am profoundly grateful for the collaboration with Ladislav Čmolík at the Czech Technical University in Prague. The partnership and his contributions were invaluable, providing essential insights that significantly enhanced the quality of work contained in this thesis. Furthermore, I extend my thanks to my colleagues at Brno University of Technology, Faculty of Information Technology, for their continuous support, stimulating discussions, and encouragement, which have enriched my experience and research. I am incredibly thankful to Veracity Protocol for being an understanding and flexible employer, granting me the necessary time and resources to pursue my doctoral studies. Their support was crucial in allowing me to balance my professional responsibilities with my academic endeavors. Lastly, I am deeply thankful to my wife and family for their unwavering support, patience, and understanding during this demanding period. Their love and encouragement have been my anchor, motivating me to persevere and achieve this significant goal.

Contents

Acronyms

2AFC Two-Alternative Forced Choice. 91, 97, 107
2IFC Two-Interval Forced Choice. 70

ANOVA Analysis of Variance. 106
ARO Active Range Optimization. 39, 40
ATC Air Traffic Control. 8

CFLM Conflict-Free Label Maximization. 14
CPU Central Processing Unit. 88
CS Clustering Search. 19

DDE Discrete Differential Evolution. 19, 20
DDEGA Discrete Differential Evolution and Genetic Algorithm. 19
DOF Degrees of Freedom. 44
DRL Deep Reinforcement Learning. 74, 76, 77, 81

EPTAS Efficient Polynomial-Time Approximation Scheme. 40

GIS Geographic Information Systems. 96, 109, 137
GLD Global Label Density. 102, 103
GPU Graphics Processing Unit. 23, 34, 87, 88
GRASP Greedy Randomized Adaptive Search Procedure. 22

HIT Human Intelligence Task. 99

ILP Integer Linear Programming. 15, 18, 19, 37

LBM Leader Bend Minimization. 31
LCM Label Conflict Minimization. 14
LCrM Leader Crossing Minimization. 31
LiDAR Light Detection And Ranging. 79
LLD Local Label Density. 103
LLDF Local Label Density Final. 103
LLM Leader Length Minimization. 31
LNM Label Number Maximization. 12, 19, 31, 39
LOM Label Overlap Minimization. 14
LOS Label Occlusion over Salience. 45
LP Linear Programming. 15
LSM Label Size Maximization. 12, 40

MADRL Multi-Agent Deep Reinforcement Learning. 77, 83, 91

MARL Multi-Agent Reinforcement Learning. 76, 77
MDP Markov Decision Process. 76, 77
MILP Mixed-Integer Linear Programming. 56, 60, 61
MIP Mixed-Integer Programming. 60, 72
MIQP Mixed-Integer Quadratic Programming. 58, 60, 61
MIS Maximum Independence Set. 16, 18, 26, 41
MLE Maximum Likelihood Estimate. 99
MMWAR Maximizing Minimum Weighted Active Range. 40
MPI Message Passing Interface. 19

PBL Particle-Based Labeling. 23, 87
PBL-A Particle-Based Adjacent Labeling. 87–93
PBL-AD Particle-Based Adjacent and Distant Labeling. 87–93
PFLP Point-Feature Label Placement. 11, 14–20, 49, 77, 78, 83, 95
POMDP Partially Observable Markov Decision Process. 77
POPMUSIC Partial Optimization Metaheuristic under Special Intensification Conditions. 17, 27
PPO Position Priority Order. 47–49, 95–97, 100–103, 105–112, 137, 143–147, 149–152
PrPO Proximal Policy Optimization. 81, 82, 88, 131

QGIS Quantum Geographic Information System. 26, 48
QP Quadratic Programming. 57, 61

RAPL Rapid Labels. 23, 87–93
RCL Restricted Candidate List. 22
RFL Reinforced Labels. 84, 85, 87–93, 113
RFL-random Untrained Reinforced Labels. 87–90
RL Reinforcement Learning. 74–77, 79, 91–93

SA Simulated Annealing. 15, 42
SAM Saliency Attention Model. 45
SARL Single-Agent Reinforcement Learning. 76, 77

WLNM Weighted Label Number Maximization. 12, 31

Chapter 1

Introduction

In today's data-driven world, visualizations play a pivotal role in transforming complex data into understandable insights, thereby facilitating effective communication and informed decision-making. These visual representations are omnipresent across various domains, including cartography, medical imaging, emergency response, and many other fields where precise and clear information presence is critical. At the heart of effective visualization lies the use of labels, which serve to annotate features with essential information such as names, measurements, or explanations, enhancing the viewer's understanding of the underlying data.

The challenge of label placement, or *labeling* in short, arises from the need to balance functional clarity with aesthetic appeal. Labels must be positioned in a way that avoids ambiguity and overlap, ensuring they are easily readable and accurately associated with their corresponding features. As the underlying data grows in complexity and density, the manual placement of labels becomes impractical, labor-intensive, and prone to inconsistencies, thus highlighting the necessity for automated solutions. This necessity drives the research focus of this thesis: the development of novel methods for automatic label placement.

Automatic label placement involves algorithmically determining the optimal positions for labels in visualizations to maintain readability and clarity without causing overlaps or visual clutter. Despite significant advancements in visualization techniques, the problem of automatic label placement remains a challenging task due to its inherent computational complexity. Specifically, achieving optimal label placement is an $\mathcal{NP}$-hard problem, requiring sophisticated heuristic and optimization methods to find feasible solutions within reasonable timeframes. Moreover, dynamic environments, such as interactive applications or real-time tracking systems, add another layer of complexity, demanding label placement techniques to adapt to data changes while maintaining clarity and coherence.

The effectiveness of label placement is also influenced by how users perceive and interact with visual information. Perceptual considerations, such as the placement within a visual hierarchy and the label's alignment with corresponding features, are crucial for optimizing user experience and comprehension. These perceptual insights can guide the development of label placement techniques that prioritize user experience, ensuring labels enhance understanding without sacrificing visual appeal.

By addressing these multifaceted challenges, this research aims to advance the field of automatic label placement, leading to the development of more powerful and versatile visualization instruments.

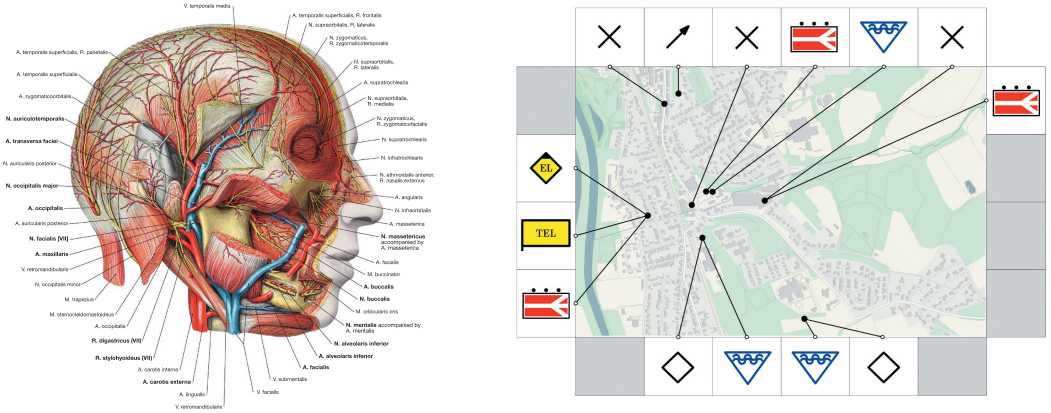
Significance of Automated Label Placement

The applications of automatic label placement are vast and varied, playing a crucial role in many fields where accurate and efficient information presentation is essential. Labels enhance visual comprehension and support critical decision-making processes across several domains.

Medical Imaging. Automatic labeling aids in annotating anatomical structures, which is crucial for diagnostics, treatment planning, and educational purposes. Labeling in medical visualizations has a long history, traditionally seen in medical textbooks and anatomy atlases; see Figure 1.1(a). As noted by Oeltze-Jafra and Preim [134], the advent of modern medical imaging and computerized medicine has expanded the range of label placement applications. Annotations assist radiologists in diagnosing image data, recording diagnostic findings, and facilitating discussions in medical team meetings. Moreover, interactive labeling allows radiologists to add comments and measurements directly to segmented pathologic structures, streamlining the diagnostic process and improving the clarity of medical visualizations. In their research, Niedermann *et al.* [133] interviewed two illustrator artists of the well-known Sobotta Atlas of Human Anatomy [137], which has approximately 1200 figures across 384 pages. The artists mentioned that label placement is a mechanical but extensively time-consuming task mainly done by hand. It takes around two hours to create a double-page layout for anatomy books. These facts demonstrate the potential time-saving benefits of automated labeling solutions. Automated label placement in medical imaging can reduce the time and effort required for manual annotation but also enhance the precision and consistency of medical documentation and contribute to more effective medical education.

Emergency Response. Emergency response services rely heavily on real-time geographical information to make critical decisions. Dispatchers must interpret maps that display the locations of emergency vehicles and incidents, necessitating labels that are clear, accurate, and devoid of clutter; see Figure 1.1(b). Gedicke *et al.* [58] highlights that clarity of information is essential for emergency response dispatchers, as their work requires them to make quick, life-saving decisions. They spend much of their time monitoring these maps, and the difference between a fast, accurate decision and a delayed or incorrect one can be critical. Therefore, the map views must not contain unnecessary clutter and should clearly convey the necessary information, especially in high-stress situations such as severe accidents where numerous emergency vehicles are deployed. By enhancing the speed and accuracy of information presentation, automated label placement helps save valuable time and human resources during emergency operations, potentially contributing to more effective incident planning and response strategies.

Cartography. In cartography, labels are essential for accurately identifying geographical features on maps, where precision and non-overlapping labels are crucial to prevent misinterpretation; see Figure 1.2(a). Manual placement of labels is a labor-intensive process that can consume a significant portion of map production time – sometimes up to 50% or more [185]. Human cartographers typically place about 20 to 30 labels per hour [42], highlighting the inefficiency of manual methods. Despite half a century of research and thousands of publications, a recent study by Harrie *et al.* [68] notes that the automation level for label placement during map production remains low. The study provides insight



(a) Vessels and nerves of the head from the well-known Sobotta Atlas of Human Anatomy [137].

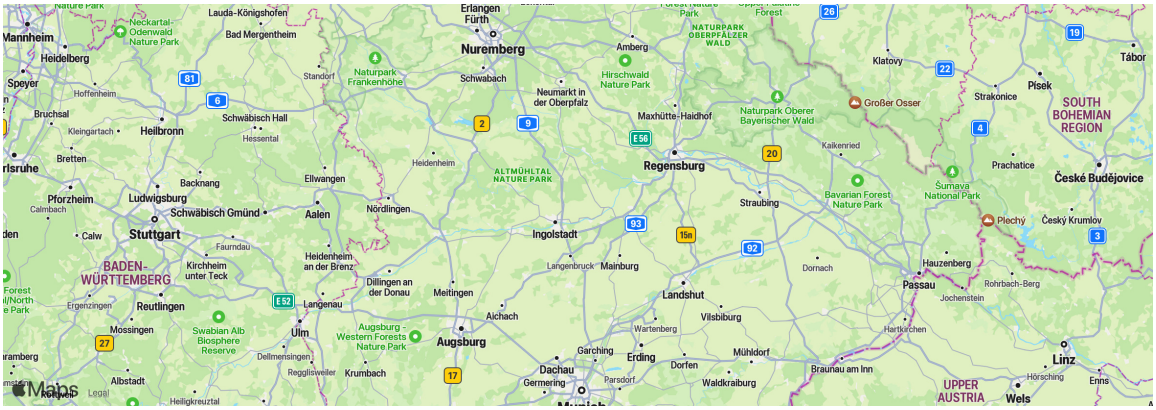
(b) Emergency response map that creates situational awareness and provide crucial information to emergency responders [58].

Figure 1.1: Label placement in medical education and emergency planning: (a) lateral view of vessels and nerves in the head and (b) emergency response map.

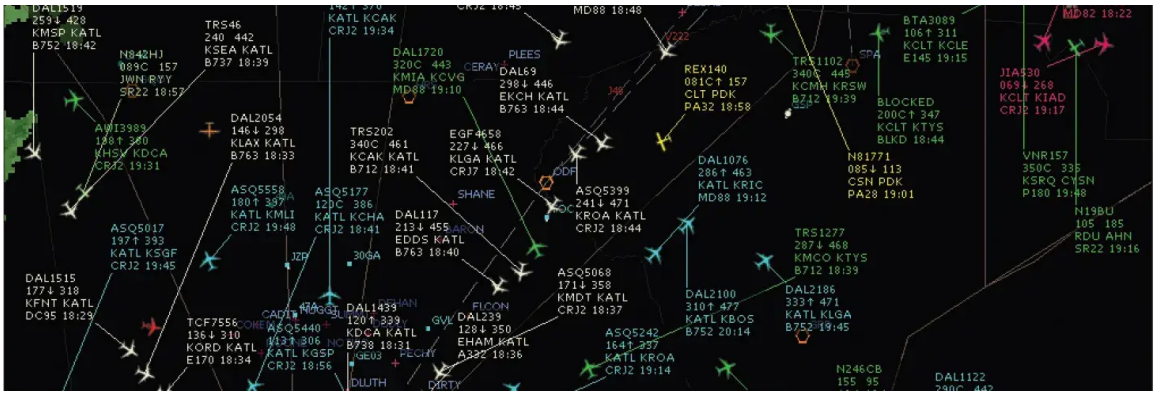
into T-Kartor, a mapping company, where an entirely satisfactory labeling solution has not been found despite efforts to increase automation. Cartographers at T-Kartor often begin with a labeling solution where all labels, including overlaps, are present and then rely on manual adjustments to resolve issues. Effective automation of label placement in cartography can address these bottlenecks, improving efficiency, reducing the workload, and freeing cartographers to focus on more creative and analytical tasks.

Air Traffic Control. In air traffic control, labels convey crucial information about aircraft, such as altitude, velocity, and identification; see Figure 1.2(b). Due to air traffic safety regulations, these labels must be present to ensure controllers have continuous access to critical data for quick decision-making [76]. Moreover, the complexity of label placement in air traffic control increases as aircraft dynamically move through the airspace. For instance, in high-density environments where numerous aircraft are tracked simultaneously, maintaining clear and distinct labels is necessary to prevent errors and ensure quick comprehension. However, the effectiveness of these labels can be compromised by overlaps. Higashikawa *et al.* [76] emphasize that minimizing label overlaps is essential because any overlap can obscure critical information, forcing controllers to manually rearrange labels to maintain readability. Such a task is time-consuming and poses a significant obstacle to efficient operations, potentially impacting safety and decision-making in high-pressure environments. Automated label placement enhances air traffic operations by ensuring labels remain clear and uncluttered. By optimizing label positions and minimizing overlaps, automated systems allow controllers to focus on managing air traffic rather than adjusting labels. This approach reduces the cognitive load on controllers and enhances situational awareness, ultimately contributing to safer and more efficient air traffic management.

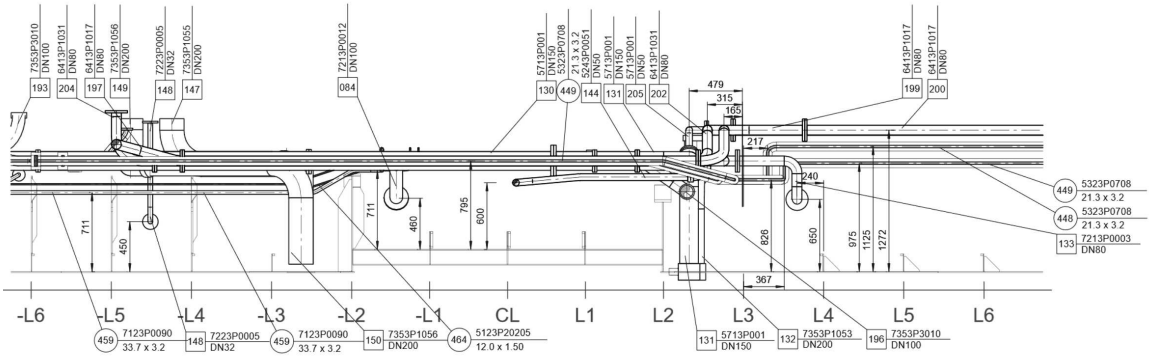
Technical Drawings. In mechanical engineering and maintenance manuals, labels are crucial in annotating technical drawings; see Figure 1.2(c). These labels provide essential information about components and their functions, aiding engineers and technicians in



(a) Map of Central Europe generated by MapKit JS provided by Apple [7], showcasing an intricate network of cities and natural reserves. The map illustrates the importance of precise and non-overlapping labels in cartography to ensure clear and accurate reading.



(b) Air traffic control (ATC) radar and onboard radar beacon transponders work together to display aircraft information on a radar screen. Each aircraft is represented by a symbol indicating direction and status. Information shown includes call sign, altitude, speed, origin, destination, and type. Symbols indicate if the aircraft is climbing or descending. Different colors represent arrivals, departures, and traffic to nearby airports [164].



(c) Technical drawing illustrating a complex piping assembly, manually annotated with labels to identify component parts and dimensions, as discussed by Lehtinen [101].

Figure 1.2: Examples of label placement in various domains: (a) cartographic label placement in a map, (b) air traffic control radar visualization, and (c) technical drawings with detailed annotations.

(a) Women's double sculls rowing broadcast during Olympics Games Paris 2024 [52].

(b) Basketball games with dynamic labels featured in the ESPN App for NBA playoffs [135].

Figure 1.3: Examples of dynamic label placement in sports visualizations: (a) rowing competition and (b) basketball game.

understanding complex assemblies. Accurate and efficient labeling in technical drawings is vital for facilitating clear communication, ensuring safety, and supporting maintenance processes. The complexity of technical drawings often requires annotating intricate details, where clarity and precision are paramount. Despite advancements in software tools, manual labeling remains prevalent in the industry. This process is time-consuming and susceptible to human error, particularly in large-scale industrial projects like shipbuilding and process plant design, where labeling can take thousands of hours [101]. As highlighted by Lehtinen [101], many existing labeling tools in technical drawing software lack the sophistication needed to tackle the challenges presented by complex assemblies and high-density drawings. These tools often fail to optimize label placement, resulting in overlaps and ambiguities that require extensive manual adjustments. Like previous applications, automated label placement can reduce the time and effort spent on manual labeling, improve consistency, and lead to more efficient design and maintenance workflows, potentially contributing to higher safety standards. Clear and precise labeling can ensure that critical safety information is easily accessible, which can prevent misunderstandings or mistakes during manufacturing, assembly, or maintenance.

Sports Visualization. In sports events such as basketball, football, car racing, or rowing, dynamic label placement is essential for displaying real-time information about participants; see Figure 1.3. These labels provide viewers with interesting details such as position, speed, and rankings, enhancing the spectator experience by delivering exciting and relevant updates. For instance, during a basketball match, labels might highlight the player currently possessing the ball and their recent performance metrics. Dynamic automated label placement offers real-time updates crucial for maintaining viewer engagement. By providing context-aware information, dynamic labeling systems create a more engaging experience for viewers and enhance their understanding of the event.

Scope and Research Objectives

The scope of this research primarily focuses on point-feature label placement, which refers to the placement of labels for distinct, often small, objects or locations within a visualization, such as cities on a map or specific data points in a graph. The research objectives aim to

advance the field of automatic label placement by developing novel methods that address the challenges associated with both internal and external labeling. Internal label placement refers to placing labels close to the point features they describe within the visualization, ensuring immediate association with their corresponding features. In contrast, external labeling involves placing labels outside the main visualization area, connected to the relevant features by lines, which is especially useful in dense visualizations to avoid clutter.

Specifically, the objectives are as follows. First, the research seeks to develop novel techniques that ensure temporally stable and visually coherent external label placements, particularly in dynamic and interactive visualizations. Second, explore the relevance of machine learning in internal label placement and propose an approach to enhance the completeness of labeled data and computation efficiency.

Third, investigate the perceptual aspects of label positioning to optimize user experience and comprehension.

Contributions and Structure

The contributions of this thesis are as follows. Chapter 2 provides a comprehensive literature review, detailing existing techniques and identifying gaps in the field of automatic label placement. Chapter 3 introduces novel approaches to temporally stable optimization of boundary labeling, presenting both offline and interactive methods. Chapter 4 explores the application of deep reinforcement learning for internal point-feature labeling, detailing the methodology and evaluation results. Chapter 5 examines perceptual prioritization in label positioning, offering insights into user preferences and optimizing placement strategies. Chapter 6 concludes the thesis by summarizing key findings, discussing limitations, and proposing directions for future research.

Chapter 2

Literature Review

At its core, label placement involves positioning textual or symbolic labels in a way that optimally augments features within visualization, such as maps or charts, without causing conflict or confusion. A *conflict* occurs when two or more labels overlap with each other or interfere with other essential elements in the visualization, obstructing their visibility and making the labels difficult to read. Effective label placement aims to minimize these conflicts to ensure readability and usability.

Automatic label[1] placement is a broad topic consisting of many different variations. The most general classification of label placement techniques is based on the position of the label relative to its associated feature. In *internal labeling*, the labels are placed closely next to the features within the visualization space itself. Common applications include maps and charts where labels must be immediately associated with geographic or data points (e.g., cities or data markers) without extending beyond the visualization's boundaries. In *external labeling*, the labels are placed outside the visualization boundary. This approach is often seen in medical atlases, where anatomical labels are placed outside the illustration, or in panoramic views of city or mountain skylines. Combining both internal and external labeling can be beneficial in specific applications, such as technical drawings, where clarity and space optimization are crucial. Both internal and external labeling can be further categorized based on the type of features they annotate: *point features* (*e.g.*, cities or mountain peaks), *line features* (*e.g.*, roads or rivers), and *area features* (*e.g.*, countries, forests or lakes).

Given the breadth of the topic, the following text will specifically focus on point features. We review internal and external labeling concerning the point-feature label placement (PFLP), publications regarding dynamic label placement, and the label placement guidelines.

2.1 Internal Labeling

Internal label placement involves positioning labels within the visualization space, close to their corresponding features; see Figure 2.1. The field of internal labeling techniques closely relates to the centuries-old discipline of cartography. Cartography, traditionally known as both an art and a science, involves the creation and study of maps. In the early 1950s, cartography was primarily a manual process, where cartographers meticulously drew maps by hand, often incorporating aesthetic elements to enhance readability and visual

[1] In the older literature also denoted as *name* placement.

(a) (b)

Figure 2.1: Examples of internal label placement in (a) web-based maps generated by MapKit JS provided by Apple [7] and (b) visualization of multi-scale and multi-instance 3D biological environments proposed by Kouřil *et al.* [97].

appeal [147]. The manual process required a deep understanding of geography, graphic design, and spatial relationships, blending technical skills with artistic creativity.

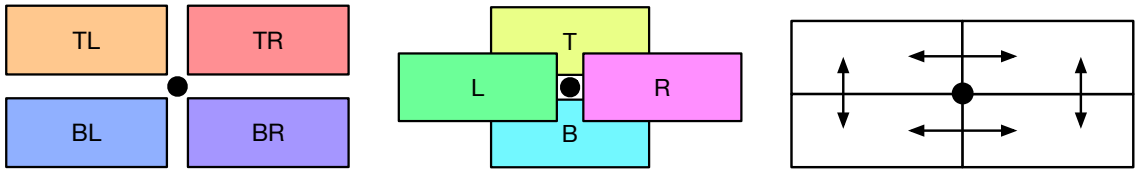
The paradigm of cartography began to shift in the 1970s with the beginning of digital technologies. A significant milestone in this transition was the work of Pinhas Yoeli [185], an Israeli cartographer, who in 1972 introduced and published the first automated map lettering system. Yoeli's system represented a significant shift from manual to automated processes, utilizing computer algorithms to place labels to maps. Yoeli's pioneering work catalyzed a wave of research and development in automated cartography. Throughout the late 20th century, researchers worldwide began exploring various methods to enhance and automate the label placement process. The proposed methods included the development of semi-automated systems, where human operators would still play a role in overseeing and adjusting the placement of labels, as well as fully automated systems that could independently determine optimal label positions based on predefined criteria.

During more than a half-decade-long research, two branches of internal label placement were formed. The *fixed-position model* works with a set of fixed candidates for each feature; see Figure 2.2(a)-(b). *Candidate*, or *candidate position*, refers to the predefined potential location where a label can be placed relative to its feature. On the other hand, in the *slider model*, the label can slide along the defined direction; see Figure 2.2(c). Table 2.1 provides a comprehensive overview of internal point-feature label placement methods and their key properties across different objectives and techniques.

Objectives. Several optimization criteria have emerged in the course of research on automated label placement, each aiming to address different aspects. The most prevalent objective is *label number maximization* (LNM), where the goal is to place as many labels with defined dimensions as possible without conflicts. Therefore, several labels can be omitted in the label placement solution. This formulation can be extended to *weighted label number maximization* (WLNM) that also considers weights of features, where the labels of features with large weight should not be missing in the solution. In contrast, features with lower weight can be removed. An alternative objective that is less common is *label size maximization* (LSM) to determine the scale factor that makes labels as large as possible while all features must be labeled without conflict [48, 54, 94, 170, 186]. Another objective, coming from air-traffic safety regulations, which require labels for all airplanes at all times,

	Zoraster [190]	Zoraster [192]	Christensen and Marks [39]	Edmondson *et al.* [51]	Zoraster [193]	Verweij and Aardal [169]	Yamamoto *et al.* [183]	Alvim *et al.* [6]	Mauri *et al.* [123]	Rylov and Reimer [148]	Haunert and Wolff [72]	Marín and Pelegrín [122]	Rabello *et al.* [143]	Lu *et al.* [114]	Higashikawa *et al.* [76]	Lessani *et al.* [102]	Ahn [3]	Freeman [55]	Cook and Jones [42, 86]	Doerschler and Freeman [49]	Ebinger and Goulette [50]	Yamamoto *et al.* [182]	Mote [128]	Cravo *et al.* [43]	Kittivorawong *et al.* [92]	Pavlovec and Čmolík [138]	Singh *et al.* [156]	Formann and Wagner *et al.* [54]	Wagner and Wolff [170]	Agarwal *et al.* [1]	Wagner *et al.* [171]	Hirsh [77]	Doddi *et al.* [48]	Kreveld *et al.* [120]	Petzold *et al.* [140]	Li *et al.* [107]	Klau and Mutzel [94]	Nascimento and Eades [130]	Klute *et al.* [96]	Luboschik *et al.* [115]	Čmolík *et al.* [41]	Kouril *et al.* [97]	Lhuillier *et al.* [105]
Feature Type																																									
point	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×
line				×										×		×	×	×	×																×				×			
area				×										×		×	×	×	×																×							
Label Shape													n/a									n/a		n/a																		
square						×																						×	×													
fixed-sized rectangle								×	×														×						x (h)			×	x (h)								
arbitrary rectangle	×	×	×	×	×		×			×	×	×		×	×	×	×	×	×	×	×				×	×			×	×	×			×	×	×	×	×	×	×	×
Objective																	n/a	n/a	n/a	n/a	n/a																		n/a	n/a	n/a
label number maximization	×	×			×	×				×		×										×			×	×	×			×	×			×							
weighted label number maximization					×						×												×									×									
label size maximization																												×	×												
conflict-free label maximization			×					×	×				×									×									×									
label conflict minimization			×	×			×	×				×		×	x (area)	×								×																
Cartographic Considerations																																						n/a	n/a	n/a
position priorities	×	×	×	×	×		×	×	×	×	×		×	×	×	×	×	×	×		×		×	×		×					×	×		×	×		×	×	×	
other factors										×	×	×		×		×	×	×		×						×														
Position Model																	n/a																					n/a	n/a	n/a
4-position			×			×	×	×	×		×		×		×				×		×	×	×	×				×	×	×	×							×	×	×	
8-position	×	×								×	×	×	×	×					×						×	×	×		×	×								×	×	×	
> 8-position				x (17)	x (24)						×					x (24)		× (6)	x (20)	×																			×	×	×	
slider																																	×	×	×				×			
Temporal Coherence / Dynamic																																								×	
Technique																																									
mathematical programming	×	×				×			×		×	×			×																										
dynamic programming																														×											
simulated annealing			×	×	×					×																									×						
genetic algorithm														×		×																									
gradient descent			×							×																															×
force-based																															×										
rule-based																	×	×	×	×																					
greedy			×							×											×	×	×	×	×	×	×												×		
metaheuristic							×	×					×											×									×					×			
other																												×	×	×	×		×	×			×	×			

Table 2.1: Overview of properties for internal point-feature label placement methods. The table summarizes key attributes of various methods used for internal label placement, including feature type, label shape, objectives, and techniques. The table also lists position models, indicating the number of candidate positions in brackets where relevant, and highlights methods optimized for temporal coherence or multiscale placement. Additional methods related to temporal coherence and dynamic placement are discussed further in Section 2.3.1 and Table 2.3. Entries with „n/a“ denote cases where information is either not applicable or unavailable. The entry „× (h)“ indicates that only the height of a label is fixed, and the width is arbitrary.



(a) (b) (c)

Figure 2.2: An illustration of the most commonly used label position models. The abbreviations of positions are as follows: top right (TR), top left (TL), bottom right (BR), bottom left (BL), top (T), left (L), bottom (B), and right (R). (a) shows the 4-position model, (b) shows the extended positions in the 8-position model, and (c) shows the slider model. The slider model can be further categorized into 1- (slides along the top of the feature), 2- (slides along the top and bottom of the feature), and 4- (slides along all possible positions of the feature) slider models.

is *conflict-free label maximization* (CFLM). In this case, the conflicts among the labels are allowed, but the number of conflicts-free labels is maximized while all features must be labeled [24,123,143]. Finally, *label conflict minimization* (LCM) also, alternatively denoted as *label overlap minimization* (LOM), appeared in the label placement research. In this problem, the conflicts among the labels are allowed, but the number of conflicts is minimized while all features must be labeled [6,76,144,145]. Interestingly, research shows that the difference between LCM and CFLM often leads to similar outcomes, as noted by Christensen and Marks [39]. Moreover, in a counterintuitive discovery, Ribeiro and Lorena [145] and Cravo *et al.* [43] observed that LCM outperformed the CFLM objective in maximizing the number of conflict-free labels.

Complexity. The point-feature label placement problem is characterized by the exponential combination explosion, computed as $candidates^{features}$, indicating the vast number of possible label configurations as the number of features and candidate positions increases. To understand the computational challenges associated with PFLP, it is essential to delve into computational complexity theory. This field helps classify problems based on the resources needed to solve them. The class $\mathcal{NP}$ (nondeterministic polynomial time) includes decision problems for which a given solution can be verified in polynomial time by a deterministic Turing machine. Within this context, $\mathcal{NP}$-completeness pertains to decision problems, such as determining if a label placement exists without conflicts. If a decision problem is $\mathcal{NP}$-complete, it indicates that the problem is as hard as any problem in $\mathcal{NP}$. On the other hand, $\mathcal{NP}$-hardness involves problems that are at least as hard as the hardest problems in $\mathcal{NP}$, typically optimization problems like maximizing the number of placed labels without conflicts. Therefore, if a decision problem is $\mathcal{NP}$-complete, then the corresponding optimization problem is $\mathcal{NP}$-hard [39]. The early research has shown that even simplified versions of the PFLP with unit-square labels are $\mathcal{NP}$-hard for most objectives, as established in various studies [54, 88, 94, 95, 121, 171]. For a small number of labels, exact algorithms exist, but they are computationally expensive and impractical for large-scale problems [94,96,169]. Consequently, heuristics are essential for finding optimal label positions in real-world scenarios. Common heuristic techniques include mathematical programming, dynamic programming, genetic algorithms, and greedy approaches, which will be elaborated on in the following section.

2.1.1 Fixed-position Model

Fixed-position models, a major category within internal label placement, restrict label positions to a predefined set of candidates relative to each feature; see Figure 2.2(a)-(b). The predefined set of positions considerably narrows the search space, making the problem more tractable while still presenting significant challenges that are $\mathcal{NP}$-hard. In the following sections, we categorize and review the related work on internal label placement with fixed models according to the various approaches that have emerged over time to address the PFLP problem.

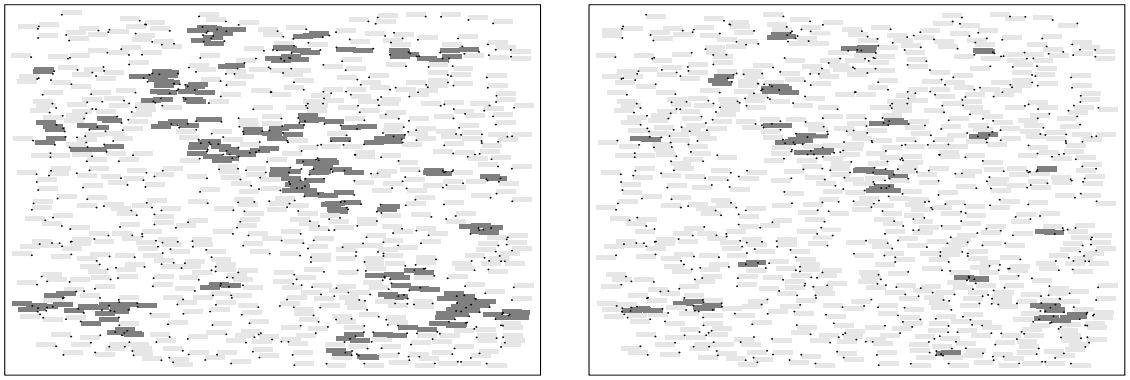
Optimization Approach

The optimization approach to point-feature label placement encompasses a range of techniques designed to find optimal or near-optimal solutions to the labeling problem. A common feature of these techniques is the mathematical formulation of an objective function, which outlines the goals for a target solution and helps differentiate which of the two solutions is superior. This approach includes mathematical programming methods, such as integer linear programming (ILP) and linear programming (LP), which provide globally optimal solutions by defining precise objective functions and constraints. While these methods ensure optimality, they require specialized solvers and can result in high computational times, limiting their practicality for large-scale applications.

In addition to mathematical programming, heuristic-based optimization methods such as genetic algorithms, simulated annealing, and tabu search offer faster and more flexible alternatives. These techniques utilize iterative processes and probabilistic rules to explore the solution space, often delivering high-quality solutions within reasonable timeframes. Genetic algorithms, for instance, apply evolutionary principles to iteratively refine label placements, balancing exploration and exploitation of the search space. Simulated annealing (SA) mimics the cooling process of metals, allowing the system to escape local optima and potentially discover a global minimum. Tabu search employs memory structures to guide the search process and avoid cycles, effectively navigating complex solution spaces. While not guaranteeing global optimality, these heuristic optimization methods offer practical solutions for real-world scenarios where computational efficiency is crucial. They are particularly valuable for applications where approximate solutions are acceptable and computation time is a constraint.

An example of mathematical programming is the work of Zoraster [190, 192], who proposed an integer linear programming formulation of label placement in maps specific to the oil industry. The objective function is defined as the sum of position penalty given by weights of individual candidates and indicator variable (denotes whether the label is placed at a given position or not) with respect to two constraints: the label can be placed only in one candidate position, and restriction of label overlap.

Christensen and Marks [39] proposed an extensive empirical study of algorithms for PFLP and introduced two optimization methods: first based on a discrete form of gradient descent and second based on simulated annealing (SA). The discrete form of gradient descent produces a higher-quality layout by allowing the continuous change of the position of already placed labels to an alternative position. However, the optimization can be stuck in local optima. The second method, based on simulated annealing, can escape from the local optima depending on the so-called temperature that gradually decreases over the annealing process. At higher temperatures, a wider range of the space can be explored. Therefore, the surrounding regions with possibly better local minima (and even the global



Zoraster's Algorithm (219) Simulated Annealing (75)

Figure 2.3: A comparison of map of 750 features with labels placed by integer linear programming algorithm of Zoraster [190, 192] and simulated annealing of Christensen and Marks [39] (origin of this visualization). Labels shown in dark gray are in conflict with other labels or features. The number in the description presents a number of conflicts.

minimum) may be found. The objective function in their study is defined by the number of obstructed labels, with an optional extension to consider the number of removed labels. Therefore, the method allows conflicts and focuses on label conflict minimization while optionally minimizing label removals. The empirical results of their study suggest that despite the quality-time tradeoff, the discretized gradient descent, the methods of Zoraster [190] and Hirsch [77] are reasonable choices. In quality-critical cases, the best results provide a simulated annealing method; see Figure 2.3.

Edmondson *et al.* [51] extended the simulated annealing approach to include other types of features beyond points and defined a numeric quality metric $q \in [0, 1]$, where 0 corresponds to ideal label placement and 1 to a borderline case where the label placement is poor but barely acceptable. For the PFLP, the quality is defined as the number of overlaps among labels and anchors.

Zoraster [193] enhanced the simulated annealing approach by incorporating a conflict elimination mechanism, prioritizing the resolution of label-label and label-feature conflicts. The method imposes higher costs on conflicts, ensuring they are resolved before other considerations, such as label deletion or repositioning. Additionally, Zoraster expanded the fixed-position model to include external labels by introducing 24 candidate positions. These positions, including additional external labels, are located in the same directions from the point feature as in the original model but at two additional distance levels, offering greater flexibility in label placement.

Verweij and Aardal [169] proposed an exact technique for provably optimal labeling. The authors reduced the fixed-position model PFLP to finding a maximum independent set[2] (MIS) in a graph posed as linear programming and solved it using a branch-and-cut algorithm. The proposed method can label up to 800 features within a moderate computation time of 20 minutes.

[2]The maximum independent set (MIS) problem involves finding the largest set of vertices (label positions) with no edges between them, meaning no overlaps. Since the MIS problem is $\mathcal{NP}$-hard, various heuristics or optimization techniques, such as greedy algorithms, simulated annealing, or integer linear programming, are used to find solutions.

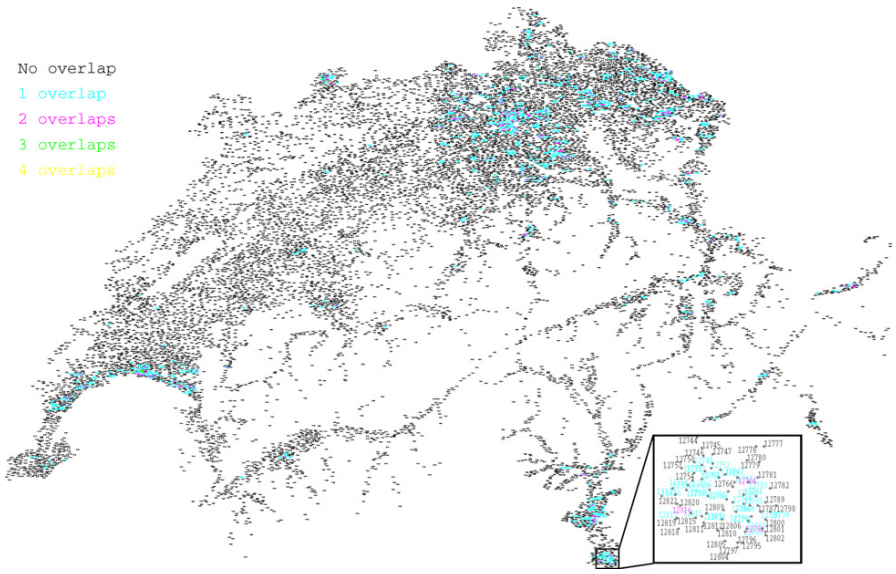


Figure 2.4: An example map of 13,206 features, identifiers of Switzerland road network, placed by technique proposed by Alvim *et al.* [6]. The number of conflicts for each label is color-coded as described in the legend in the upper left corner.

Yamamoto *et al.* [183] introduced the combinatorial optimization method based on Tabu Search, where conflicts in the final label placement are allowed. The idea is to step out of the local optima concerning the Tabu List of previously visited solution candidates within a short period. The presented study showed that Tabu Search provides better results of labeling quality, concerning a number of overlaps among labels and the priority of candidate positions, than other described methods [39, 77, 190].

Alvim *et al.* [6] proposed a point-feature labeling method to minimize the number of overlapping labels based on the application of the POPMUSIC (Partial Optimization Metaheuristic under Special Intensification Conditions) metaheuristic. The POPMUSIC metaheuristic divides the PFLP into subproblems that are optimized with Tabu Search. The framework leverages the strengths of Tabu Search for local optimization while managing large-scale problems through the partitioning mechanism of POPMUSIC. The methodology begins with a constructive procedure to generate an initial solution, which is then iteratively improved by solving smaller, overlapping subproblems. Each subproblem is optimized using Tabu Search until no further improvements can be made. The experimental results demonstrated that the POPMUSIC approach is highly efficient and significantly faster than previously published methods, achieving high-quality solutions with minimal computational effort; see Figure 2.4. However, while the technique minimizes the number of overlapping labels, it encounters challenges when addressing specific cartographic considerations, indicating room for further enhancements to integrate cartographic preferences into the optimization process fully.

Mauri *et al.* [123] proposed a new mathematical formulation and a Lagrangian decomposition technique for the PFLP. The formulation uses a 0-1 integer linear programming approach to maximize the number of conflict-free labels (*i.e.*, conflicts are allowed). Additionally, the authors presented an alternative objective focused on minimizing the number

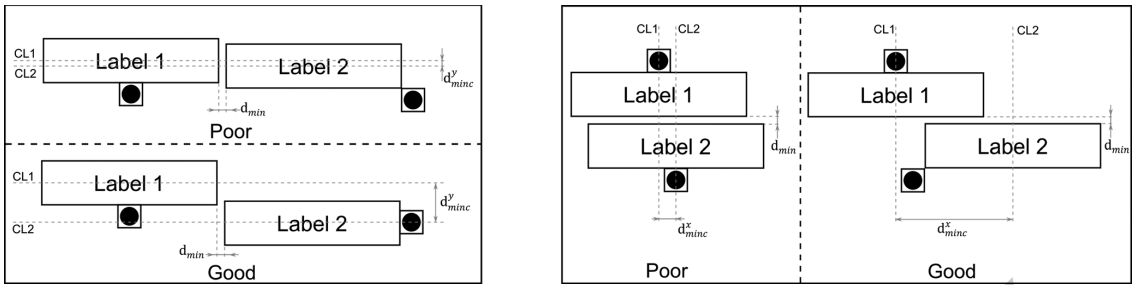


Figure 2.5: An example of ambiguity between two labels proposed by Rylov and Reimer [148]. For each label pair, the distance among label boundaries d_{min} and also the distance among horizontal and vertical centers d^x_{minc} and d^y_{minc} are combined to measure ambiguity. When the distances exceed a given threshold, the ambiguity term does not penalize the objective function.

of conflicts. The Lagrangian decomposition involves partitioning the *conflict graph*[3], into several sub-graphs (clusters). To mitigate the impact on the global optimal solution, some vertices of the conflict graph (*i.e.*, candidate positions) are copied into clusters to be fully defined globally. The authors propose to duplicate the vertices with the greatest number of inter-cluster edges. Finally, the local solutions of individual clusters are synchronized using additional constraints. The solution for 1000 labels takes, on average, over 3 hours, while the optimality gap (*i.e.*, a measure of the solution's quality compared to the optimal) is 0.48%. The solution without Lagrangean decomposition is an order of magnitude faster (2 hours) but has an optimality gap of 10.64%.

Rylov and Reimer [148] proposed mathematical formulation using an 8-position model that considers cartography rules and guidelines. The objective function is constructed as a weighted sum of simple metrics corresponding with cartographic rules. Among common PFLP requirements, such as conflict-free placement or label candidate priority, the authors include guidelines such as labels of coastal places should be placed on water bodies instead of land, and the relationship between the label and feature should be unambiguous and cluttered. The ambiguity measure is based on the Euclidean distance of each label pair considering distance among label boundaries d_{min} and also distance among horizontal and vertical centers d^x_{minc} and d^y_{minc} as shown in Figure 2.5. When the distances exceed a given threshold, the ambiguity term does not penalize the objective function. The objective function was optimized in experiments using greedy, discrete gradient descent and simulated annealing. Consistently with Christensen and Marks [39] they found that simulated annealing outperforms the other techniques in the achieved quality of label placement.

Haunert and Wolff [72] proposed ILP formulation using an arbitrary n-position model considering label-feature ambiguity and clutter, extending the formulation of Rylov and Reimer [148] that is not strictly ILP. The technique is based on *inference graph*. Contrary to the conflict graph, the inference graph contains edges for label positions that are conflict-

[3] A conflict graph, initially proposed by Ahn and Freeman [3,55] and later extended by Agarwal *et al.* [1], is a conceptual tool used to manage and resolve label overlaps in the fixed-position model. In this graph representation, each node corresponds to a candidate label position, and each edge represents a conflict between two labels, indicating that the labels overlap if placed simultaneously. The problem of selecting the maximum subset of non-overlapping labels can be formulated as a maximum independent set (MIS) problem in the conflict graph as proposed by Agarwal *et al.* [1].

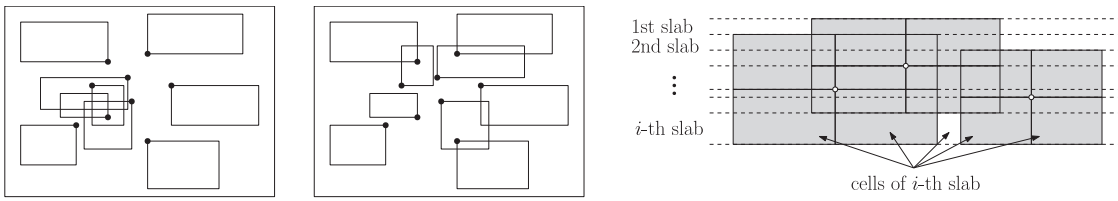
free but ambiguous. The ambiguity is expressed by edge weight in the graph, which is a user-selectable parameter. Label positions are added to the graph only when within a distance of λ is located in a feature unrelated to the label. Therefore, the relation of two label positions is considered for the ambiguity in the objective function. Moreover, contrary to common label number maximization (LNM), the authors argue that LNM is not practical as it is not the goal to completely overlap the visualization with densely packed labels, as is typically the case of LNM. Instead, they defined a shape (*e.g.*, round disk or square) that at any position in the visualization can be intersected by at the most K_τ. This allows authors to control the label density of the final solution.

Similarly, Marín and Pelegrín [122] proposed an ILP formulation that addresses ambiguity in map labeling based on the proximity of labels to unrelated features. They argue that ambiguity arises when labels are close to features they do not relate to, considering the minimum distance between the four vertices of a rectangular label and other features as a measure of label position ambiguity. Therefore, they formulated the problem using objective functions that minimize the distance-based ambiguity measure, subject to constraints ensuring label placement feasibility and conflict resolution. Unlike the former approach, they define ambiguity for a single label, not for a pair of labels. They presented two ILP formulations, each allowing for placing all labels, including those with conflicts, and modification for only placing labels without conflict. Compared to other PFLP techniques, they use an unusual 8-position model based on a typical 4-position model while extended with the same positions rotated 45 degrees counterclockwise. Additionally, they comprehensively reviewed ILP techniques for point-feature label placement.

Rabello *et al.* [143] presented a clustering search metaheuristic (CS) for the point-feature labeling with the objective to maximize the number of conflict-free labels as a new alternative to solve the PFLP problem. The CS can detect promising areas of the search space and prevent more intense searches in poor areas or areas that have already been sufficiently explored. The authors conducted a case study that shows that the percentage of conflict-free labels obtained by CS is the highest among the other compared methods. Lu *et al.* [114] proposed a hybrid approach combining discrete differential evolution and genetic labeling algorithm (DDEGA) with the objective to minimize the number of overlapping labels for all the different types of features (*i.e.*, point, line, area). Further, the conducted experiment shows that the fitness function converged more quickly than a genetic algorithm (GA) and discrete differential evolution (DDE) alone. Nevertheless, the proposed approach only uses some of the most basic rules (*e.g.*, label conflict, label-feature conflict, label non-ambiguity, and label priority).

Higashikawa *et al.* [76] proposed an integer linear programming approach for minimizing label overlap, where all features must be labeled. The goal is to find a placement of labels such that the maximum overlap at any point p in the plane is minimized; see the difference between Figure 2.6(a) and 2.6(b). The authors divide the visualization plane into cells. A cell is a smaller region of the plane formed by the intersection of horizontal and vertical lines drawn through the edges of label candidates. These lines partition the plane into horizontal and vertical strips, and the intersections of these strips create cells; see Figure 2.6(c). By using cells, instead of considering every possible point p in the plane, the problem only needs to consider the overlap within each cell. Within each cell, the overlap remains constant for all points p in that cell. Finally, they propose linear programming relaxation and rounding 4-approximation.

Lessani *et al.* [102] proposed an MPI-based parallel genetic algorithm for the placement of multiple geographical feature labels with the objective of label conflict minimization.



(a) Free-label maximization (b) Overlap minimization (c) Division cells

Figure 2.6: Examples of optimal solutions of (a) free-label maximization, (a) overlap minimization and (c) division of the visualization plane into cells proposed by Higashikawa *et al.* [76]. Using cells simplifies the problem as it only needs to consider the overlap within each cell rather than evaluating every possible point p in the plane.

The approach addresses point, line, and area features. Point features have 24 candidate label positions across three distance levels from the feature. The quality function includes binary detection for label-label conflicts, a constant value for label-feature conflicts, and a normalized distance from the center of the label to the feature for label association. The optimization uses a combination of genetic algorithms and discrete differential evolution (DDE) based on Lu *et al.* [114]. Post-optimization, a refinement step slides labels along four directions (in an X-shape), with up to 20 attempts per conflicted label, further reducing label-feature conflicts. Experimental results show improvements in both label placement quality and computational efficiency.

Gedicke *et al.* [59] investigated labeling methods for zoomless maps specifically designed for small-screen devices such as smartwatches. They proposed an approach that distributes labels over multiple pages, allowing users to navigate through pages without overlapping labels or the need for zooming. The approach involves a two-phase strategy: a pre-processing phase that uses an optimization approach to pre-compute labelings at the city level, called global labelings, and a query phase for on-demand retrieval of individual labelings at a more local level, called local labelings; see Figure 2.7. The optimization criteria include minimizing the number of pages, prioritizing important labels to appear earlier, and ensuring a spatially balanced distribution of labels. The authors used an integer linear programming formulation to solve the problem, treating it as a graph coloring problem. The number of restaurants contained in the cities' experiment frames ranges from 2070 in Calgary to 5349 in Las Vegas. Most instances were solved within a few minutes up to one hour at maximum.

Exhaustive (Rule-based) Search Approach

The exhaustive search approach to point-feature label placement systematically explores all possible configurations to find an optimal solution, leveraging predefined rules to guide the search process. This approach performs well in small-scale problems where the search space remains manageable. However, the exponential growth of the search space limits the practicality of exhaustive search in larger instances of the PFLP problem.

Ahn and Freeman [3, 55] proposed a graph-based approach, where a node represents a label and nodes are connected by an edge if the corresponding labels overlap. The graph, nowadays commonly called a conflict graph, is divided into individual connected components (a graph that contains only nodes that can be reached mutually by traversing the edges)

Figure 2.7: An example of a small-screen device approach consisting of three global labeling pages with an illustration of two local labelings (A, B) proposed by Gedicke *et al.* [59]. Time-consuming global labelings are pre-computed during a preprocessing phase, done once, and then local labelings are extracted during a query phase. In the query phase, local labeling is achieved by sampling all labels contained in the frame representing the current region of interest for each page.

that can be processed individually, as a node in one component can not affect the label in another. The placement process employs a modified A* algorithm, prioritizing nodes with the smallest degree of freedom to minimize backtracking. If it becomes impossible to place a label, the algorithm backtracks, aided by update records that track changes in free-space blocks and node degrees of freedom.

Cook and Jones [42, 86] proposed a rule-based algorithm for label placement using the logic programming language Prolog. The strategy for resolving label conflicts involves a systematic search for possible positions, utilizing several heuristics. Labels are processed in descending order of importance. Labels are placed in decreasing order of placement difficulty, starting with those having fewer options in dense areas. Among labels with equal placement prospects, those with the most potential overlaps are placed first. If a label cannot be placed, the algorithm backtracks to previously placed labels to find an alternative position until another candidate position is available. If no such position exists, the label that caused the backtracking is removed.

Doerschler and Freeman [49] introduced a rule-based algorithm for dense-map label placement. The authors build upon the backtracking approach but extend it using the quality-evaluation rule, which provides a normalized value $q \in [0, 1]$. When the quality exceeds the threshold, the other candidate positions are evaluated, or backtracking occurs. The threshold is initially set to 1 and gradually decreases with each rule evaluation, meaning that initially unacceptable placements may become acceptable if better placements cannot be found easily. After a label is placed, the quality of neighboring labels is reevaluated, potentially resulting in their repositioning.

Greedy Approach

The greedy approach to point-feature label placement offers a straightforward and efficient method for addressing the labeling problem. Unlike exhaustive or optimization techniques, which aim to explore a larger portion of the solution space or guarantee global optimality, greedy methods focus on making locally optimal choices at each step. This approach involves applying algorithms like the sweep-line algorithm to sequentially label point features based on immediate, short-term criteria. Greedy algorithms evaluate potential label positions and select the best option available at each stage without considering the broader implications or allowing backtracking to correct previous choices. As a result, while greedy methods may not always find the optimal overall solution, they can produce a satisfactory label layout with significantly reduced computational time, often in milliseconds. This speed advantage makes greedy algorithms particularly suitable for time-critical applications such as real-time mapping and dynamic visualization tasks. However, the absence of backtracking or limited backtracking depth means that greedy methods may miss more optimal configurations, especially in densely packed or complex label placement scenarios. Consequently, while these methods are efficient and practical, they typically result in fewer labels being placed compared to former approaches.

Ebinger and Goulette [50] proposed a simple greedy approach for label placement for the 1990 United States Decennial Census maps. The method sequentially attempts to place the label at the candidate positions defined by the 4-position model. If the label cannot be placed at any of these candidate positions, only the anchor is retained, and the label is discarded. Among the optimization techniques, Christensen and Marks [39] also proposed a greedy-based method where the label is placed in the best available position considering all previously placed labels. The position of already placed labels can not be changed. Consequently, the label is hidden if no position is available.

Yamamoto *et al.* [182] proposed an algorithm for point-feature label placement, where conflicts in the final label placement are allowed, for real-time screen maps, aiming to balance label placement quality and processing time. The algorithm uses a precomputed conflict graph to identify overlapping label positions and applies a three-step approach: maximum non-conflict labeling, conflict minimization for remaining labels, and local search for further refinement. Each node in the conflict graph represents a candidate position, and the edge denotes a conflict with the other node, *i.e.*, candidate position. In the first step, the algorithm iteratively places the label with the smallest number of conflicts. Later, all unsolved labels are placed in the position with the least conflicts. Finally, local search is applied to labels that are in conflict, and the modification of the candidate position is accepted if the number of conflicts is reduced. Mote [128] introduced a greedy method that divides the screen into a 2-dimensional grid and determines a simplified version of a conflict graph of the point features. Only point features in neighboring cells need to be checked for conflict with a given point feature. The technique assigns a cost to every label candidate to find the label layout and selects the least expensive set of non-conflicting candidates.

Cravo *et al.* [43] proposed a greedy randomized adaptive search procedure (GRASP), a metaheuristic consisting of two phases: a constructive phase and a local search phase. In the constructive phase, a feasible solution is iteratively built by blending greedy and random selection methods to form a restricted candidate list (RCL) of label positions. The vertex degrees of the conflict graph, representing potential overlaps, are used to order the candidate positions. The solution is constructed by iteratively selecting and placing the candidate position with the least conflicts from the RCL. The local search phase explores

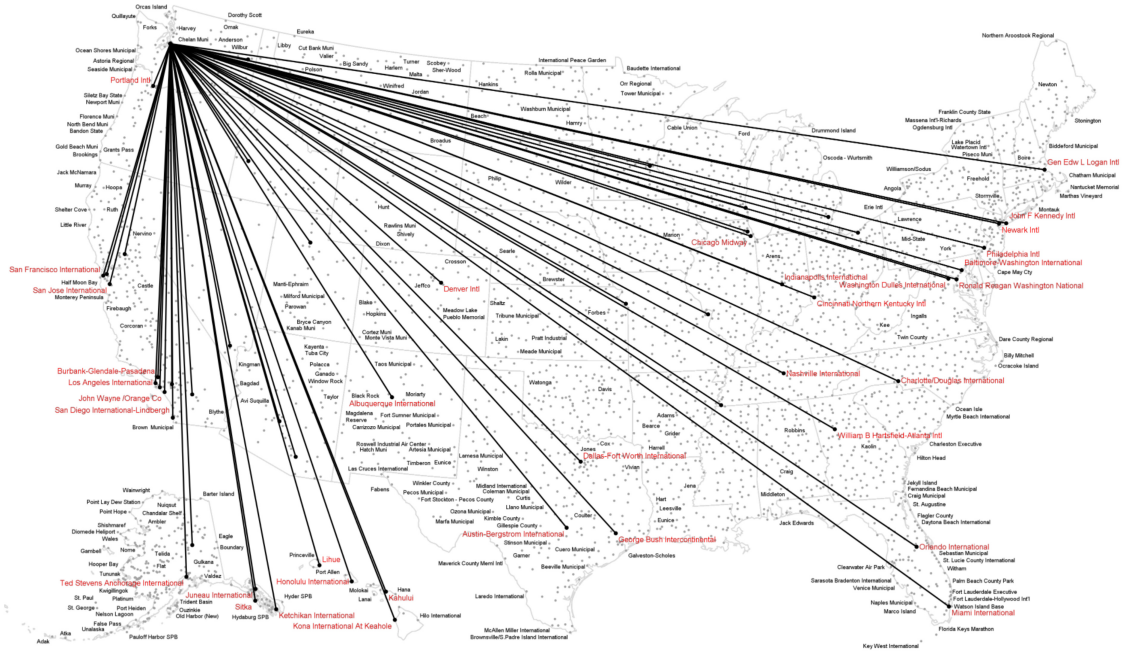


Figure 2.8: An example of US airports and direct flights to Seattle-Tacoma International Airport (denoted by red labels) labeled by the method proposed by Pavlovec and Čmolík [138]. Red labels are not allowed to intersect lines and features. Labels denoted by black color are not allowed to intersect lines, features, or boundaries of US states.

the neighborhood of the current solution, attempting to improve it by reassigning candidate positions to reduce conflicts. The algorithm checks each feature for alternative valid candidate positions and accepts changes that improve the solution.

Kittivorawong *et al.* [92] addressed the performance problem of particle-based labeling (PBL) proposed by Luboschik *et al.* [115] (see in Section 2.1.3) with an occupancy bitmask that allows faster evaluation of label overlaps with complex visual features that cannot be occluded. Nevertheless, the computation time required to determine overlap for a label still depends on the size of the label and the resolution of the screen. Pavlovec and Čmolík [138] introduced Rapid labels (RAPL), an approach leveraging the power of GPU. They allow labeling several point features in each iteration using a 2-dimensional grid. Further, they evaluate the overlaps of labels with important visual elements, the conflicts between the labels, and the ambiguity of the labels to position the labels in suitable order; see Figure 2.8. They utilize Summed Area Table to evaluate the overlaps, conflicts, and ambiguity independently of label size and screen resolution.

Singh *et al.* [156] introduced a greedy heuristic that separates the 8-position model into two 4-position models (TR, TL, BL, BR and T, L, B, R); see Figure 2.2. The solution involves creating a conflict graph of candidate labels and finding an independent set through geometric sweep-line methods in different directions (top-to-bottom, left-to-right, bottom-to-top, right-to-left). Subsequently, they merged the solutions from these directions using a bipartite graph approach to maximize labeled points.

Formal Algorithm Approach

The point-feature label placement problem has also attracted significant attention from the formal algorithm community. Researchers in this domain focus on developing algorithms that provide theoretical guarantees of solution quality and computational efficiency. These formal algorithms often aim to find solutions that are provably close to optimal, typically within a factor f of the best possible outcome (*e.g.*, $f = \frac{1}{2}$ indicates the solution is at least half as good as the optimal one). Formal algorithms are characterized by rigorous proofs of runtime complexity, which provide insights into their performance and scalability. However, these algorithms often rely on simplifying assumptions to manage complexity and facilitate analysis. For instance, many formal approaches assume uniform label dimensions, such as unit square rectangles or labels with consistent height.

For instance, Formann and Wagner *et al.* [54] proved that maximizing the size of uniform axis-parallel square labels (*i.e.*, all labels having same dimensions which is highly uncommon in geographical maps, but can be the case of technical maps as denoted by authors), where n is the number or anchors while using the 4-position model for candidates, is $\mathcal{NP}$-complete, and proposed an $\mathcal{O}(n \log n)$ approximation. Among others, the authors also proved that no polynomial-time approximation algorithm can exceed a guarantee factor $f = \frac{1}{2}$. Later, Wagner [172] proved that the same problem has $\Omega(n \log n)$ lower bound on the approximation runtime.

Wagner and Wolff [170] claimed that the previous algorithm is the best possible from a theoretical point of view but is useless in practice as it typically produces non-optimal solutions that are not much better than 50% of the optimum. To address this, they proposed a heuristic for label size maximization of uniform axis-parallel square labels with the same guarantee and complexity, but the practical results are closer to the optimum. The results were compared with random and real-world samples where the exact optimum is known. Agarwal *et al.* [1] proposed an $\mathcal{O}(n \log n)$-time factor-$\frac{1}{2}$ approximation based on divide-and-conquer algorithm and dynamic programming[4] for unit height and varying width labels label number maximization.

Wagner *et al.* [171], similar to Kakoulis and Tollis [87], proposed a two-phase approach for arbitrary axis-aligned labels. In the first phase, the number of possible candidates is reduced by three rules while the optimality (*i.e.*, maximum number of labels that can be placed) is not reduced. In the second phase, the authors reintroduced a conflict graph, initially proposed by Ahn and Freeman [3, 55], to detect conflicts and effectively reduce the number of candidates for each feature to one. The idea of the algorithm is to start with features that have the maximum number of candidates. For each feature, the candidate with the maximum number of conflicts (*i.e.*, number of edges) is removed. The authors show that the proposed method is faster than simulated annealing but performs slightly worse regarding a number of labeled features.

2.1.2 Slider Model

Unlike fixed-position models, slider models provide greater label placement flexibility by allowing continuous label movement around their corresponding features. Consequently, the

[4]Dynamic programming is an optimization strategy that simplifies complex problems by dividing them into smaller, manageable subproblems. Each subproblem is solved only once, and its solution is stored for future reference. This approach eliminates redundant computations and is particularly practical for problems characterized by overlapping subproblems and an optimal substructure, where the best solution can be derived from the best solutions of the subproblems.

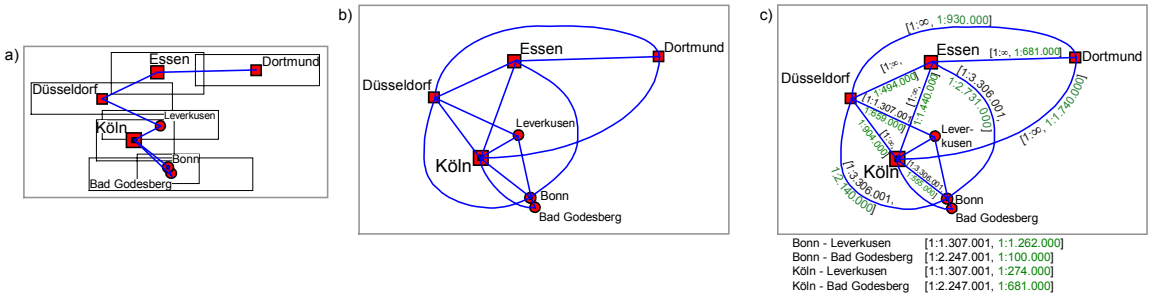


Figure 2.9: An example of conflict graph types used in technique proposed by Petzold *et al.* [140]. (a) Static conflict graph for a given scale. (b) Scale-independent conflict graph extended by the attributed edges with a scale interval (c).

slider model offers the potential for reducing overlaps and improving readability. However, slider models have not received much attention from the scientific community, and only a few approaches have been proposed.

In particular, Hirsh [77] proposed a forced-based approach[5] for conflict-free label maximization where the translation vectors are repeatedly computed based on repelling forces for overlapping labels. Doddi *et al.* [48] introduced a slider model, where, like Formann and Wagner *et al.* [54], the goal is to maximize the size of equally-sized labels that can be rectangular or elliptical. Moreover, the presented method allows the rotation of each label around the corresponding feature. Finally, they also introduce a bi-criteria algorithm that allows to keep some of the labels unlabeled.

Kreveld *et al.* [120] studied several sliding models (1-, 2- and 4-slider model) and proposed a combinatorial $\mathcal{O}(n \log n)$-time greedy factor-$\frac{1}{2}$ approximation approach for all mentioned slider models with the label number maximization objective. They assume fixed size height but flexible width of labels. They also proved that the 4-slider model is $\mathcal{NP}$-complete. The authors also shown an interesting fact, that proposed greedy 4-slider algorithm can place about 10-15% more labels than corresponding 4-position model. Surprisingly, in comparison to the fixed-position model approach by Christensen and Marks [39], the proposed 4-slider approximation performs slightly worse up until 750 features, and from that point to 1500 (the maximum number of tested points), only marginally better.

Petzold *et al.* [140] introduced a fast greedy slider model approach with the label number maximization objective suitable for zooming and scrolling. The main idea is to divide the label placement process into two phases: (1) time-consuming preprocessing and (2) fast interaction phase. In the first phase, the scale-independent conflict graph is constructed, similar to the approach of Ahn and Freeman [3, 55], extended by the attributed edges with a scale interval. The lower/upper bound is the smallest/largest scale where a conflict exists. Similarly, the nodes in the graph are attributed with the smallest scale, where the corresponding feature is labeled. In the second phase, the static conflict graph for a given scale and clipping rectangle is derived from the scale-independent conflict graph constructed in the preprocessing phase; see Figure 2.9. Finally, the conflict-free candidates are selected by traversing the static conflict graph in ascending priority order.

[5] A force-based approach utilizes physical metaphors of forces, such as attraction and repulsion, to determine optimal label placement. Labels and features are treated as objects that exert forces on each other. Attractive forces pull labels toward their associated features, while repulsive forces push labels away from other labels and visual elements to prevent overlap. This dynamic adjustment continues until the system reaches an equilibrium, a state where the forces are balanced.

Li *et al.* [107] proposed a method based on the region of movability that comes from plane collision detection theory. The idea is to define a conflict-free label placement search space and then derive the best position for each label using heuristic search methods. Nevertheless, the method does not contain the removal strategy for cases where the features are too densely distributed; therefore, conflict-free search space does not exist in such a case.

2.1.3 Other Models

The exploration of label placement models beyond the traditional fixed-position and slider models has led to innovative approaches that enhance flexibility and user interaction in the label placement process. These alternative models often integrate interactive elements, human-in-the-loop techniques, and hybrid strategies combining internal and external labeling methods.

In particular, Klau and Mutzel [94] proposed an exact provably optimal exponential-time approach for label number maximization formulated as zero-one integer linear programming. The proposed approach is independent of the labeling model and can be used both for fixed-position and slider models. Moreover, the approach can place several labels of arbitrary size per feature, which is rare in regard to the previous work, and also allows the combination of several labeling models for different sets of features. The main idea of the approach is to decompose the problem into a horizontal and vertical component and link these by additional constraints. The authors also show that the more flexible the labeling model is, the easier is the computation. The provided evaluation shows that the exponential explosion is at about 400 labels for the 4-position and 4-slider models. Interestingly, the computation time of the method is highly dependent on the spatial configuration features, as for a specific instance of 366 features, while using the 4-position model, it takes more than two and a half days to find an optimal solution. Nascimento and Eades [130] proposed an iterative framework that allows users to interact with the labeling process by providing domain knowledge (hints) via a conflict graph to aid the placement method. The idea of hints is to help the algorithm escape from local minima or reduce the solution space. These hints can include font size changes, custom candidate positions, modification of the cost of candidate positions, and modifications of edges in the conflict graph. The framework was evaluated by professional cartographers, who provided positive feedback on its potential utility. They appreciated the integration of human expertise with automatic optimization methods, although they noted the need for further enhancements and additional features for practical deployment.

Klute *et al.* [96] proposed another approach that integrates the human-in-the-loop concept into label placement algorithms. Their method begins with an algorithmically generated initial solution, which a cartographer can refine through various modifications, such as changing label positions or sizes, deleting labels, or fixing label candidates; see Figure 2.10. The primary goal is to maintain the stability of the label placement after modifications, ensuring that most labels retain their positions. The authors developed a user interface that allows cartographers to make these adjustments easily. They implemented a range of label placement algorithms based on conflict graphs, including greedy approaches, several approximations of the maximum independence set (MIS), and exact MIS techniques. Additionally, they incorporated three algorithms from QGIS. The evaluation involved simulating modifications through five random rounds of resizing and deletion operations. The criteria for evaluation included labeling quality, time complexity, and stability of modifications.

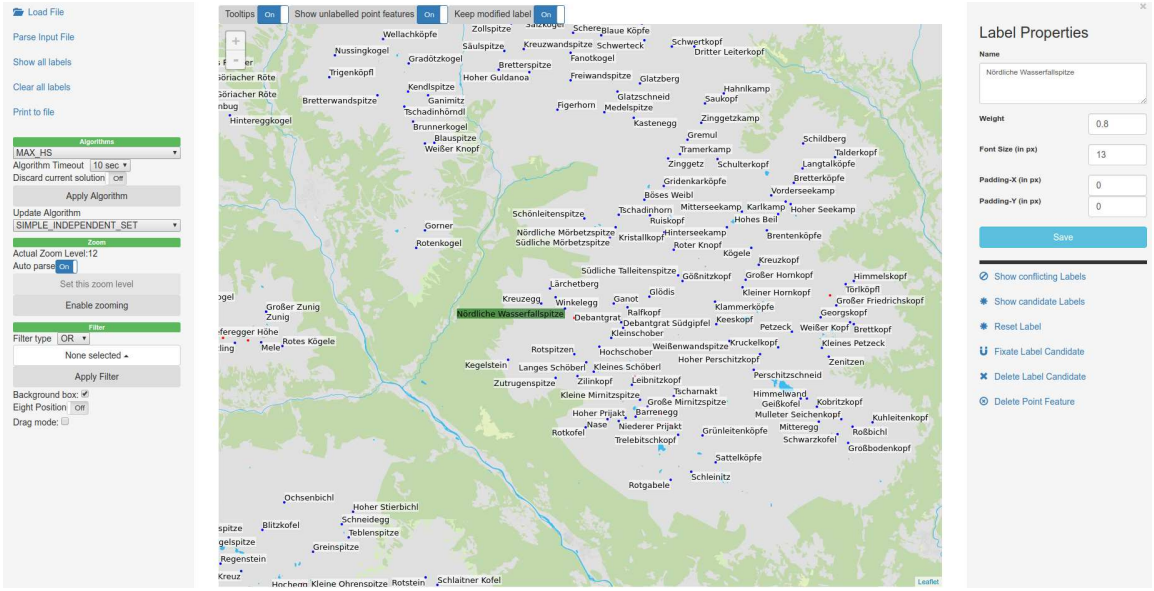


Figure 2.10: An example of the graphical user interface proposed by Klute *et al.* [96]. Users can refine the algorithmically generated initial solution by changing label positions or sizes, deleting labels, or fixing label candidates.

The results showed that a combination of the initial solution via the KaMIS solver [99] and modifications using the POPMUSIC approach [6] yielded the most promising outcomes. The experiments also highlighted that the density of labels significantly affects the runtime, ranging from 120ms for 182 sparsely distributed features to one hour for 373 densely distributed features using the KaMIS solver.

Luboschik *et al.* [115] proposed a greedy particle-based labeling approach that can also respect other visual elements and the visual extent of labeled features. The visualization space is discretized into two types of so-called *conflict particles*. The *label particles* represents point features, and *virtual particles* represents the space occupied by other visual elements. The placed label is approximated by a set of newly generated virtual particles. Candidate positions for label placement are determined sequentially using several well-known techniques in order: (1) 4-position model, (2) 8-position model, (3) 4-slider model, (4) proposed spiral model. The candidate positions are first determined by the 4-position model, and if a position is feasible (*i.e.*, no conflict particle is in the area of the label), the label is placed, and the following set of candidates is not generated. However, if the candidate positions are not feasible, another set of candidate positions is generated by the 8-position model, and so on. The last candidate positions for labels are generated using introduced *spiral model* – candidate positions are sampled from a spiral function. Furthermore, more distant features are connected with the label by a straight leader line (*i.e.*, using external labels). The authors suggest that the proposed approach enables interactive labeling of up to 1000 features.

A similar approach combining internal and external labels proposed Čmolík *et al.* [41]; see Figure 2.11. The technique operates in screen space, making it suitable for 2D and 3D projected visualizations. The position and type of each label are specified by an ambiguity threshold, which can be set by the user. The algorithm takes a color buffer and an ID buffer as input to represent unique parts of the visualization. The labeling criteria are

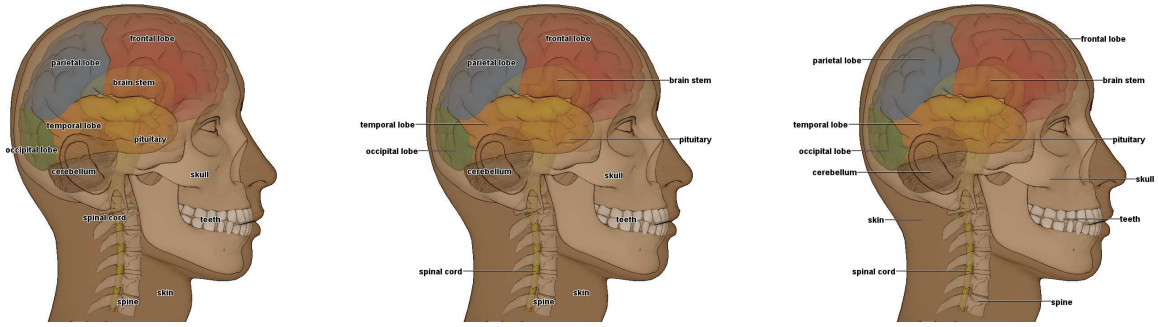
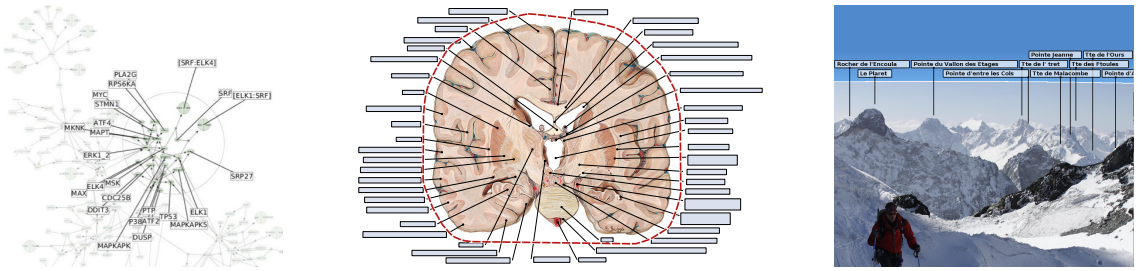


Figure 2.11: An example of a human head model with placed labels by technique proposed by Čmolík *et al.* [41]. The image illustrates the changes in label layout when changing the ambiguity threshold from zero (a) over midpoint (b) to the maximum value (c). Notice the ambiguous spinal cord and spine placement in (b) in contrast with (c).

described by a fitness function comprising salience, priority of label candidates, leader line length, and area ambiguity. Labels are placed using a greedy algorithm, which iteratively selects the best candidate label based on the fitness function that aggregates the above criteria. The use of an ambiguity threshold allows the algorithm to switch to external labels when internal labels would cause confusion, ensuring that all labels are clearly associated with their respective objects. Although the technique allows for real-time usage, temporal coherency is not addressed in this work.

Kouřil *et al.* [97] introduced a multi-scale and multi-level internal label placement framework for labeling area features approximated by point features based on greedy approximation; see Figure 2.1(b). The method works as a post-processing step; nevertheless, the scene is supposed to be rendered with the approach of Le Muzic *et al.* [100]. The input is a set of buffers containing information about object type and id, depth, color, and a scene graph. The method consists of three stages. First, the scene is divided into three regions (foreground, middle-ground, and background) based on the distance of the object to the camera (the idea is to group objects further from the camera). Afterward, the representative instance for each object type is determined using multi-criteria fuzzy decision-making. Several criteria are considered, such as salience – to label the most prominent instances and temporal coherence – to steer the algorithm toward the previous position. Finally, the labels are centered on the computed anchor points of the area features, using 3D billboards that are aligned with the camera. However, because points approximate area features, they can overlap, especially for longer texts, which reduces their readability. Additionally, the authors acknowledge that the current solution has the limitation of labels potentially occluding the objects they are intended to describe, which further impacts the clarity of the visualization.

Lhuillier *et al.* [105] proposed a density-based labeling approach to place labels into uncluttered parts of underlying visualization. The idea is to create a density map of the features represented in the 2D space by convolving with a radial density estimation kernel. Afterward, the constructed density map, in combination with gradient descent, is used to guide the label from the feature point toward the area of low density. The gradient descent trajectory is then used as a leader line in case of more distant labels from the feature. Unfortunately, the method does not consider any well-established guidelines for



(a) (b) (c)

Figure 2.12: (a) a part of a biological network with the focus region [75], (b) an visualization of labels around the contour of a brain [133], (c) an application of boundary labeling for visualization of mountain peaks [29].

label placement. Furthermore, the leaders of features on a gradient descent trajectory are merged into a single leader. Therefore, the relation of labels with the corresponding features remains unclearin this case.

2.2 External Labeling

As already mentioned, labels play a crucial role in identifying and describing features, but placing labels internally within the visualization space can lead to clutter and overlaps, especially when dealing with densely packed data or complex visual structures. Therefore, to overcome the shortcoming of internal label placement, *external* label placement techniques emerged. External labels are placed outside the primary visualization area with the ambition of reducing clutter and improving readability. A key component of external labeling is the use of *leaders*, which are lines that connect labels to their respective features. Leaders can vary in shape and type, such as straight, curved, or angular, helping to maintain clear associations between labels and features while minimizing visual clutter.

The term external label placement is widely accepted in the current literature as an umbrella term encompassing several labeling techniques. Such as *excentric* or *focus-region* labeling [53], which focuses on labeling features within a specific region of interest (typically described by a circle of fixed radius), often referred to as a focus region or lens; see Figure 2.12(a). Another technique is *contour* labeling [4], which restricts label placement to a predefined contour that generally matches the shape of the illustration; see Figure 2.12(b). Lastly, *boundary* labeling [16], and its variant for panorama images [29, 60], involves placing labels along the edges of a rectangular boundary surrounding the visualization; see Figure 2.12(c).

The topic of external label placement is extensively covered in the state-of-the-art report by Bekos *et al.* [20,21], which provides the first unified taxonomy for categorizing the various external labeling techniques and presents a comprehensive survey of their applications and developments. Therefore, the following text is structured as a high-level overview of external labeling techniques, summarizing key aspects and advancements. Table 2.2 provides a comprehensive overview of external point-feature label placement methods and their key properties across different objectives and techniques.

	Fekete and Plaisant [53]	Bertini *et al.* [25]	Heinsohn *et al.* [75]	Niedermann and Haunert [132]	Hartmann *et al.* [69]	Ali *et al.* [4]	Čmolík and Bittner [40]	Niedermann *et al.* [133]	Bekos *et al.* [17, 18]	Bekos *et al.* [16]	Benkert *et al.* [23]	Bekos *et al.* [19]	Lin *et al.* [111]	Lin *et al.* [112]	Lin [110]	Maass and Döllner [117]	Gemsa *et al.* [60]	Huang *et al.* [81]	Kindermann *et al.* [90]	Kindermann *et al.* [91]	Bose *et al.* [31]	Bose *et al.* [32]	Jia *et al.* [85]	Gedicke *et al.* [57]	Gedicke *et al.* [58]	Preim *et al.* [141]	Azuma and Furmanski *et al.* [9]	Mühler and Preim [129]	Götzelmann *et al.* [65]	Götzelmann *et al.* [66]	Stein and Décoret [159]	Sirk *et al.* [157]	Tanzgern *et al.* [163]	Hegle *et al.* [74]
Feature Type																																		
point	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×			×	×	×	
area						×	×																×					×	×	×				×
Label Shape																																		
circle				×																														
fixed-sized rectangle						×			×	×	×	×	×		×			×		×			×	×	×	×	×	×	×		×	×	×	
arbitrary rectangle	×	×	×		×		×	×	×			×		×		×		×	×	×		×	×	×	×	×	×	×	×		×	×		×
Leader Type																																		
o-leader																×	×																	
p-leader							×		×	×																								
po-leader						×	×		×	×	×		×					×	×	×	×	×		×										
opo-leader									×	×			×	×	×			×	×		×													
do-/od-leader	×	×									×	×																						
s-leader	×		×	×	×	×	×	×	×														×		×	×	×	×	×	×	×	×	×	×
other		×																																
Objective	n/a	n/a														n/a	n/a					n/a								n/a				n/a
label number maximization						×																												
label size maximization										×																								
label conflict minimization				×																			×			×	×		×					
leader crossing minimization			×																				×					×	×		×	×		
leader length minimization					×	×	×	×	×		×	×	×	×				×	×	×	×		×	×	×			×	×		×	×	×	
leader bend minimization									×									×			×													
Contour																											n/a							
1-sided rectangle						×	×	×	×	×	×	×	×	×	×	×	×	×	×				×	×										
2-sided rectangle						×	×	×	×		×	×	×	×	×			×	×	×	×	×				×								
3-sided rectangle																				×	×	×												
4-sided rectangle									×			×	×							×	×	×			×									
convex hull					×	×	×	×																										
other hull																												×	×	×	×	×		
circle	×	×	×	×	×	×	×	×															×	×	×									
Temporal Coherence / Dynamic	×	×	×	×		×	×									×										×	×	×	×	×	×	×	×	×
Technique																																		
mathematical programming				×													×							×	×									
dynamic programming								×		×	×			×			×	×	×	×	×	×												
simulated annealing																									×		×							
force-based			×	×	×	×																						×					×	
greedy							×						×				×						×			×	×		×	×	×	×		×
metaheuristic														×	×	×								×										
other	×	×	×			×			×			×	×								×			×	×		×							

Table 2.2: Overview of properties for external point-feature label placement methods. The table summarizes key attributes of various methods used for external label placement, including feature type, label shape, leader types, and objectives. The table also lists contour and technique types where relevant and highlights methods optimized for temporal coherence or multiscale placement as discussed further in Section 2.3.2. Entries with „n/a“ denote cases where information is either not applicable or unavailable. The table is partially adapted from Bekos *et al.* [21].

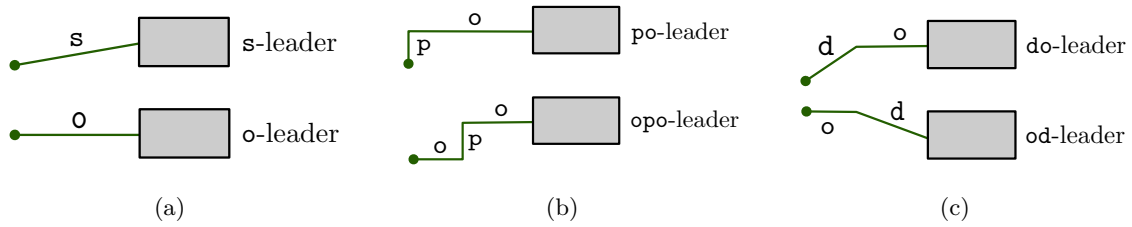


Figure 2.13: (a) straight-type leaders, (b) orthogonal-type leaders, (c) diagonal-type leaders. The segments are defined with respect to the reference edge that is the left edge of a label.

Leader types. Systematic classification of different leader types was introduced by Bekos *et al.* [16] and further extended in their subsequent work [20, 21]. Leader line λ consists of k segments that are in order from feature to label. The type of i-th segment is described by character c from $T = \{\mathtt{s}, \mathtt{r}, \mathtt{d}, \mathtt{o}, \mathtt{p}\}$, thus the shape of λ is described by string $z \in T^k$. The description of each character $c \in T$ is following:

`s` is a straight line segment,
`r` is a ray segment emitted from a center,
`d` is a diagonal line segment,
`o` is an orthogonal line segment,
`p` is a parallel line segment.

The `o`- and `p`- segments are defined with respect to the reference edge, which is typically the vertical edge of a label. The leaders consisting only of those segments are called *orthogonal*. The leaders comprising `d` in combination with preceding or succeeding `o`- and `p`- segments are called *octilinear*. See Figure 2.13 for an illustration of common leader types.

Objectives. During the external label placement research, several optimization criteria emerged. Similar to internal label placement, *label number maximization* (LNM) aims to maximize the number of placed labels in visualizations while the leader crossings are not allowed. This objective is intended for visualizations with many labels where all labels cannot fit the visualization space. The extension of this case is *weighted label number maximization* (WLNM), which also takes into account the weight of individual features. Another objective is *leader crossing minimization* (LCrM), where the objective is to minimize the number of intersections among leaders as it makes the feature-label assignment ambiguous and the visualization cluttered. However, the most common and explored objective is *leader length minimization* (LLM), where the goal is to minimize the length of leaders as long leaders make the feature-label assignment time-consuming, cluttered, and ambiguous. Alternative reasoning is the ink savings when these visualizations are printed. Finally, *leader bend minimization* (LBM) emerged in external label placement. The objective is to minimize the number of leader bends as it makes the visualization cluttered and hard to read.

Complexity. When dealing with labels of varying dimensions, external label placement problems are typically classified as $\mathcal{NP}$-hard [21]. This complexity arises from optimizing label positions and routing leader lines, which connect labels to their corresponding features. Specifically, objectives such as leader length minimization [19, 112] and leader crossing minimization [15, 110] have been proven to be $\mathcal{NP}$-hard in multiple studies. Additionally, the work of Gemsa *et al.* [63] and Bekos *et al.* [18] demonstrates that both panorama

labeling and 2-opposite-sided boundary labeling also fall into the $\mathcal{NP}$-hard category. This classification indicates that finding optimal solutions for these problems is computationally challenging, similar to internal label placement. Despite the complexity, heuristic methods such as force-based techniques, dynamic programming, and mathematical programming are crucial for achieving optimal or near-optimal solutions in practical applications.

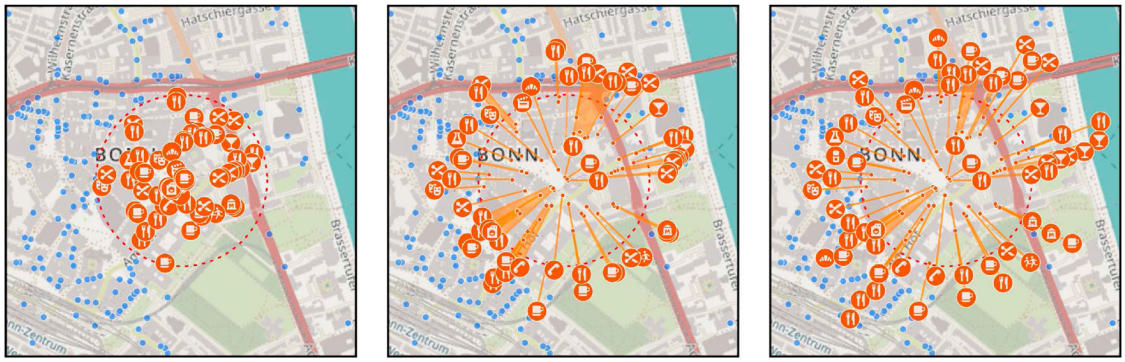
2.2.1 Excentric Labeling

Excentric labeling is a technique used to manage the presentation of labels in visualizations by dynamically displaying labels for only the most relevant features within a specific area of interest. This method is beneficial for densely populated visualizations where displaying all labels simultaneously would lead to clutter and reduced readability.

The concept of excentric labeling was introduced by Fekete and Plaisant [53]. The authors describe their method as dynamic in a way that labels for features are implicitly hidden, and when the cursor stays more than one second over a circularly defined focus area, the labels for the features within the defined area are populated. The labels are stacked along the left and right sides of the focus area and connected with features by `do`-leaders (left side) and `s`-leaders (right side). When a vertical or horizontal ordering of the features is important, the algorithm allows the crossing of the leaders. Further, their method was extended by Bertini *et al.* [25] work that improves the previous limitations by addressing high-density areas and uneven-density distributions. They propose to create a density map such that an element of this map denotes the number of features that fall inside. Afterward, the diameter of the focus area is obtained using average pooling on the density map. Additionally, summary statistics in the form of a bar chart are provided above or below the lens, offering insights into the area's content.

Heinsohn *et al.* [75] proposed four different approaches for label placement with `s`-leaders. The first (so-called Left Hand All) approach assigns the labels only to the left side of the focus area. In the second so-called radial approach, the leader is defined by the path of a ray given by the radial projection of the feature point to the circumference of the focus area. When the angle between two subsequent leaders is too small, the corresponding labels can overlap each other. To resolve this, they shift the labels further away from the center of the focus area until there are no overlaps left. The third approach is based on a force-based approach such that the preferred position of a label is its radial position. The last so-called cake-cutting approach places the labels equally distributed along the circumference of the focus area. Overlapping labels are resolved using the same method as in the radial approach.

Niedermann and Haunert [132] proposed a focus region map labeling approach to enhance the usability of digital maps, especially on small screens, by reducing the need for zooming. The technique places labels near the surroundings of the focus region using fish-eye projection [181], ensuring that labels move smoothly without flickering when the user moves the focus region. The projection ensures that the position of a focus label stays on the radial ray from the center of the focus region through the original position of the label, preserving the radial ordering of the focus labels; see Figure 2.14. When the user stops moving the focus region, the positions of the focus labels are optimized to reduce overlaps using a force-based approach. This approach introduces repulsive forces between potentially overlapping labels and attractive forces between labels and their initial positions. Mathematical programming is used to find an equilibrium of these forces. The method also allows centric label placement for features in the middle of the focus area, considering their



(a) (b) (c)

Figure 2.14: An illustration of the technique proposed by Niedermann and Haunert [132]. (a) The input consists of labels within the focus region. (b) Phase 1: The labels are projected using fish-eye projections [181]. (c) Phase 2: The positions are refined using optimization using force-based formulation.

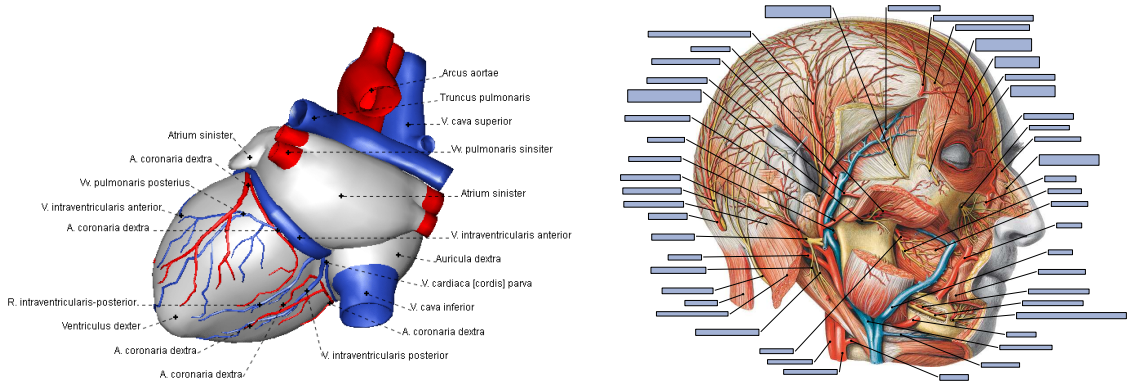
importance to the user's interests. Finally, leaders who are close to each other are bundled. Instead of drawing each leader, each bundle is presented as a slightly transparent polygon enclosing the bundled leaders.

2.2.2 Contour Labeling

Contour labeling is a technique designed to improve label organization and clarity in visualizations by aligning labels along predefined contours that follow the shape of the underlying illustration. This method is particularly effective in scenarios where it is crucial to maintain a visual flow and coherence with the visualization, such as in anatomical images.

In particular, Hartmann *et al.* [69] proposed a method for determining appealing placements of textual annotations for complex-shaped geometric models using dynamic potential fields. The technique employs artificial potential fields, where attractive forces draw labels toward their reference objects, and repulsive forces prevent overlaps with other labels, object boundaries, and image borders. Label candidates are evaluated based on various criteria, including the accumulated potential of their placement area, visibility, length of the connecting line, and the angle between the connecting line and the main axis (horizontal or vertical).

Ali *et al.* [4] analyzed hand-drawn illustrations from anatomic atlases and anatomic textbooks and presented several labeling methods based on rectangular, circular, and silhouette (convex hull with padding) contours of the illustration; see Figure 2.15(a). In the image analysis stage, the anchor points (features) of the illustration are computed using a domain-specific 2D segmentation of the illustrated object. In the rectangular 2-sided labeling method, the labels are divided into left and right (or top and bottom) parts using pivot (*i.e.*, mean or median of anchor points). Afterward, the labels are placed on the contour and connected to the feature with `s`- or `po`-leaders. In the circular/silhouette method, the position of labels is given by the radial projection of the anchor points to the contour. The possible overlap between labels is resolved iteratively using a force-based approach. Ultimately, if the leader-to-leader intersection is present, the positions of the labels are switched until all intersections are resolved. Later, Čmolík and Bittner [40] presented a



(a) (b)

Figure 2.15: An example of label placement for a medical illustration of (a) a heart proposed by Ali *et al.* [4] and (b) a head proposed by Niedermann *et al.* [133].

similar algorithm based on fuzzy logic and greedy optimization that can calculate a more reasonable set of anchors while being faster due to the implementation on GPU.

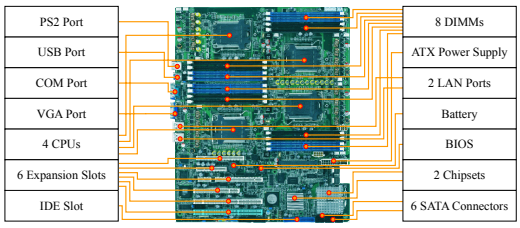
Niedermann *et al.* [133] proposed a method for labeling anatomic atlases based on a contour of the figure prescribing the possible positions of the labels; see Figure 2.15(b). The algorithm is based on dynamic programming, and the quality criteria can be defined as hard constraints that cannot be violated or as soft constraints whose violation is penalized by a general cost function. They formally proved that the resulting label layout satisfies all hard constraints and has a minimal cost.

2.2.3 Boundary Labeling

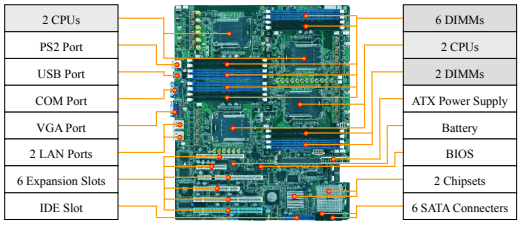
Boundary labeling is a technique used in visualizations to place labels along the edges of a rectangular boundary surrounding the illustration. This approach is beneficial for managing label placement in dense visual environments, where internal labels might cause clutter or overlap with essential visual elements. Boundary labeling is often categorized into different models based on how many sides of the boundary the labels can occupy. The specific category is typically denoted as i-sided, meaning the labels can be placed on i sides of the boundary rectangle, where $i \leq 4$.

Bekos *et al.* [16, 17] introduced a method for boundary labeling where a set of features is connected with a set of predefined labels positioned in one or up to three rows with rectilinear leaders. In the following publication, Bekos *et al.* [18] presented a method that minimizes the total leader length or, in the case of orthogonal leaders, the total number of bends. Benkert *et al.* [23] presented algorithms for solving the 1- and 2-sided boundary labeling problem for `po`- and `do`-leaders and also showed that better label layouts could be produced if one considers criteria such as the number of bends of the leaders and distance between the leaders, in addition to the length of the leaders. Later, Bekos *et al.* [19] extended the work of Benkert *et al.* [23] and published the boundary labeling method with octilinear leaders for up to 4-sided boundary labeling that minimizes the total leader length.

Lin *et al.* [111] introduced *many-to-one boundary labeling*, where more than one feature can be connected to an identical label; see Figure 2.16(a). They proposed a 1- and 2-sided boundary labeling algorithm with `opo`- and `po`-leaders such that the total number of crossings among leaders is minimized. Further, Lin *et al.* [112] introduced the concept



(a) Many-to-one boundary labeling by Lin *et al.* [111]



(b) Many-to-one boundary labeling using hyperleaders by Lin *et al.* [110]

Figure 2.16: An example of label placement for a motherboard image using many-to-one label placement proposed by Lin *et al.* [110,111]. (a) More than one feature can be connected to an identical label. (b) In addition, the *hyperleader*, may also consist of several branches connecting multiple features with a label. The duplicated labels are highlighted in gray.

of *1.5-side boundary labeling* for the text annotation systems and the polynomial-time algorithm to solve the problem such that no crossings among leaders are allowed and the total leader length is minimized. Additionally, they explored two-side boundary labeling for the annotation systems and showed that the problem is $\mathcal{NP}$-complete. To address the complexity, they proposed a heuristic approach based on a genetic algorithm. In the following work, Lin [110] presented a polynomial-time approach for removing the crossing among the leaders in many-to-one boundary labeling by so-called *hyperleaders* – leaders that consist of several branches connecting multiple features to a single label; see Figure 2.16(b). Therefore, the method allows for the duplication of labels, with the primary objective being to minimize the number of duplicate labels while also minimizing the total length of the hyperleaders. The number of bends in the leaders is addressed in a post-processing stage to optimize the label layout further.

Maass and Döllner [117] presented two real-time labeling techniques for virtual landscapes. Both techniques are iterative and process the labels by the distance of the features from the camera. Also, the labels are always centered on their leaders. The first method, denoted as *growing border*, places labels in available empty space divided vertically into horizontal slots. Once a label is placed, the area occupied by the label intersecting with the slots, as well as the space below the label, is marked as occupied. The second method, referred to as *interval slot*, operates similarly to the growing border technique, with one key difference: only the area of the label intersecting the slots is marked as occupied upon placement. Additionally, they outlined general characteristics for labeling perspective views of terrain-based scenes: (1) Labels for features closer to the observer should be positioned in the lower area of the view, while labels for features farther away should be placed in the upper area. (2) Labels near the observer tend to hold more interest than those farther away.

Gemsa *et al.* [60] presented 1-sided multirow algorithms with cross-free o-leaders for panorama images based on dynamic programming that for a set of features finds the label placement on the lowest number of rows (MinRow) or maximizes the number of labels (MaxLabels) placed in a given number of available rows K. Each label is placed above the horizontal line and connected with the corresponding feature by a vertical leader that does not intersect with any other label. They show a $\mathcal{O}(K^* n^3)$ algorithm for MinRow where K^* is the number of rows in the optimal MinRow solution and $\mathcal{O}(Kn^3)$ algorithm for MaxLabels.

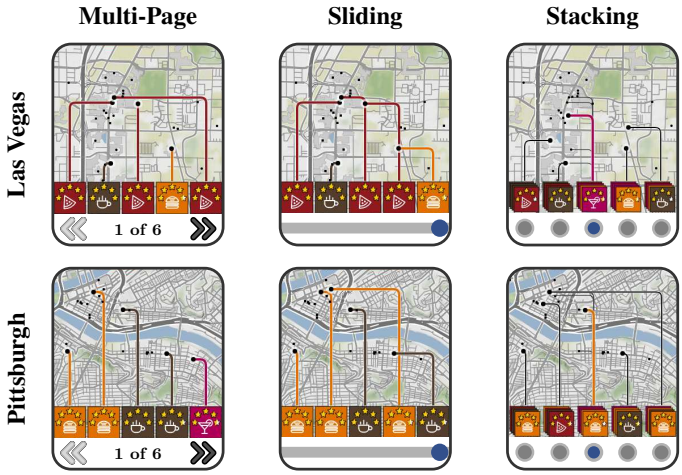


Figure 2.17: An illustration of three boundary labeling approaches suitable for small-screen devices proposed by Gedicke *et al.* [57]. The labels can be viewed through multiple pages, sliding along the bottom boundary, or listing through a stack of labels.

Huang *et al.* [81] introduced a flexible 2-sided boundary model with cross-free `po`- and `opo`-leaders where the labels are allowed to be placed at any non-overlapping position contrary to previous approaches, where the labels can be placed only into a stack of predefined equidistantly distributed positions. The proposed definition allows using `s`-leaders more often, leading to a shorter total length of leaders. The objective is either the total leader length minimization or the total bend number minimization. Simultaneously, Kindermann *et al.* [90, 91] proposed a 2-sided model with cross-free `po`-leaders running in $\mathcal{O}(n^8 \log n)$ for the total leader length is minimization and $\mathcal{O}(n^3 \log n)$ for label number maximization. Furthermore, they showed that the 3-sided model can be decomposed into two 2-sided subproblems. Bose *et al.* [31] improved a 2-sided boundary model with cross-free `po`-leaders for the total leader length minimization using dynamic programming running in $\mathcal{O}(n^3 \log n)$. Moreover, they show an algorithm for 3- and 4-sided models with `po`-leaders running in $\mathcal{O}(n^5)$. Finally, they show a 2-sided model with `po`-leaders running in $\mathcal{O}(n^6)$ and $\mathcal{O}(n^9)$ for `opo`-leaders. Later, Bose *et al.* [32] proposed 3- and 4-sided boundary model with cross-free `po`-leaders with improved running times $\mathcal{O}(n^5)$ and $\mathcal{O}(n^3 \log n)$, respectively.

Jia *et al.* [85] proposed a greedy energy-based approach for semantic-aware external label placement in augmented reality. The awareness is driven by salience map estimation, semantic segmentation (DeepLabV3), and edge detection (Canny). The authors let 20 participants place labels for 5 to 9 objects in 300 photos of street views. Given these observations, they calculated placement priors for each semantic category. For example, it is not desirable to place labels over traffic signs or lights, but it is valid for sky or foliage. In combination with a salience map, semantic segmentation, and edge detector, they create a guidance map. The label placement technique is formulated as energy minimization with an objective function consisting of overlaps of labels with guidance map and edges, label overlaps, leader line length, intersections, and orientation (vertical leaders are preferred). Finally, using a greedy algorithm, labels are sequentially placed. Note that the method is not strictly boundary labeling as it enables the placement of the labels within the visualization space.

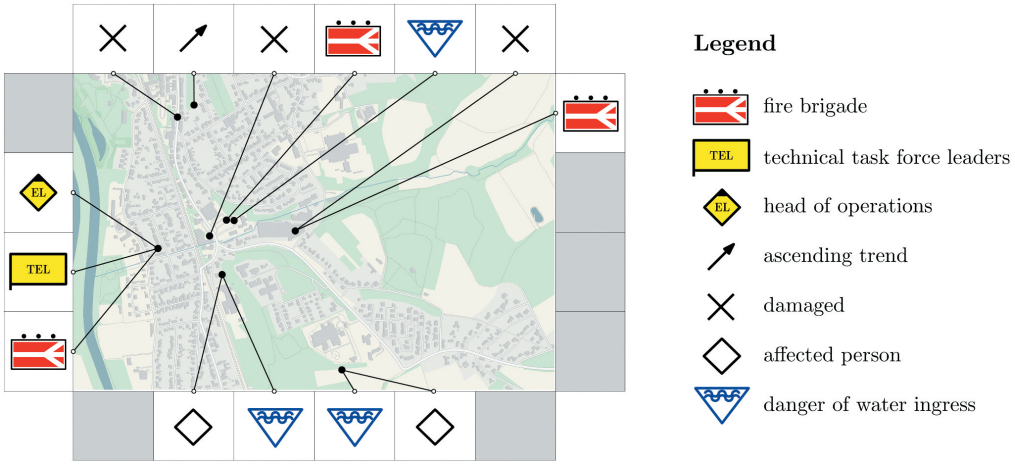


Figure 2.18: An example of situation map for emergency response proposed by Gedicke *et al.* [58]. These maps serve to create situational awareness and provide crucial mission-related information to emergency responders.

Gedicke *et al.* [57] investigated labeling methods for zoomless maps designed explicitly for small-screen devices such as smartwatches; see Figure 2.17. They distinguished three labeling methods: multi-page boundary labeling, sliding boundary labeling, and stacking boundary labeling. These methods allow users to slide labels along the bottom side of the map, browse labels based on pages, or stack labels. The authors proposed exact algorithms and fast, simple heuristics to solve the optimization problems. The optimization considers various criteria, including the ranking of labels, the total leader length, the crossings of leaders, and the distance between leaders. For multi-page boundary labeling, the problem is transformed into finding a perfect matching in a bipartite graph, seeking a minimum-weight perfect matching that minimizes the sum of the weights over all selected edges. The authors used a linear programming approach to solve this problem. The sliding boundary labeling is approached as a constrained orienteering problem. In this context, a graph-based method is used where each vertex represents a state with a score, and each edge has a length. The goal is to find a path from the source to the target vertex, maximizing the total score along the path while ensuring that a given length is not exceeded. An integer linear programming (ILP) formulation expresses the problem of finding an optimal labeling, defined as a linear objective function subject to a set of linear constraints. The stacking boundary labeling approach is based on the method by Benkert *et al.* [23]. It involves partitioning the map into strips induced by vertical lines through each label's port and feature. The problem is transformed into a static boundary labeling problem, with the objective of minimizing leader length while ensuring the labeling is crossing-free.

Later, Gedicke *et al.* [58] proposed exact and heuristic approaches for the automatic placement of tactical symbols in situation maps; see Figure 2.18. Their method automates the process based on an established map layout that distributes symbols to the map boundaries. The approach places tactical symbols such that crossing leaders are prohibited, prioritizing symbols of highly relevant features, minimizing the length of leaders, and grouping symbols with similar semantics or spatial proximity. The exact approach employs combinatorial optimization expressed as an integer linear program (ILP), while the heuristic approach is based on simulated annealing, which provides approximate solutions to opti-

	Been *et al.* [13, 14]	Liao *et al.* [109]	Zhang *et al.* [187, 188]	Gemsa *et al.* [63, 64]	Yokosuka and Imai [186]	Barth *et al.* [12]	Gemsa *et al.* [61]	Vaaraniemi *et al.* [168]	Bhore *et al.* [26]	He *et al.* [73]	Petzold *et al.* [140]	Kouřil *et al.* [97]
Feature Type												
point	×	×	×	×	×	×	×	×	×	×	×	×
line	×	×						×			×	
area	×	×						×			×	×
Label Shape												
square			×	×	×		×					
fixed-sized rectangle		×		×	×				×			
arbitrary rectangle	×	×			×	×	×			×	×	×
Objective												n/a
label number maximization	×	×	×	×						×	×	
label size maximization					×				×			
minimum weighted active range maximization			×									
active range maximization	×	×		×	×	×	×	×				
Temporal Coherence / Dynamic	×	×	×	×	×	×	×	×	×	×	×	×
Internactions									n/a	n/a		
zooming	×	×	×			×					×	×
panning						×	×	×			×	×
rotation				×	×	×	×	×				×
Technique												
mathematical programming			×	×		×	×					
dynamic programming							×					
simulated annealing										×		
force-based								×		×		
greedy	×			×		×	×		×		×	×
metaheuristic												
other	×	×	×		×				×			

Table 2.3: Overview of properties for dynamic internal point-feature label placement methods. The table summarizes key attributes of various methods, including feature type, label shape, objectives, and techniques. Entries with „n/a“ denote cases where information is either not applicable or unavailable.

mization problems. The heuristic approach generates an initial solution by transforming the labeling problem into finding a maximum weighted matching in a bipartite graph consisting of all features and ports as vertices. The heuristic produces high-quality results, often matching or exceeding the quality of optimal results for prioritizing and leader length criteria. Running the exact approach took between five seconds and ten minutes, which is impractical for interactive map design. In contrast, the heuristic generated solutions in under a second for each of the 30 instances, with running times between 0.44 and 0.82 seconds.

2.3 Dynamic Labeling

Maintaining a stable and coherent label layout is crucial in dynamic environments with interactive or continuously updated visualizations. Dynamic label placement addresses the challenge of positioning labels in a way that ensures smooth transitions and minimizes abrupt changes, thereby enhancing the readability and aesthetic quality of the visualization. This stability is essential in applications such as real-time data visualization, interactive maps, and animations, where users need to track labels and easily understand their association with dynamic features. Table 2.2 and Table 2.3 provide a comprehensive overview of dynamic external and internal point-feature label placement methods and their key properties across different objectives and techniques.

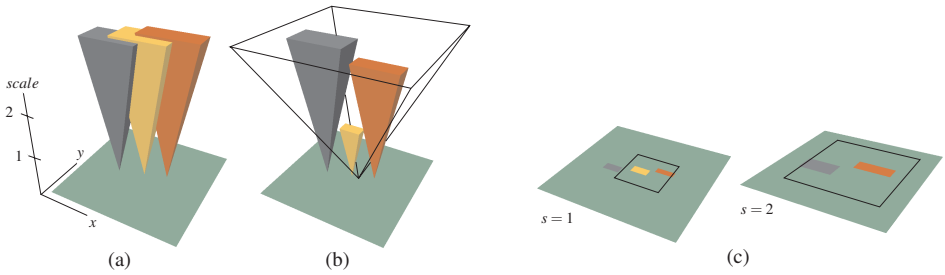
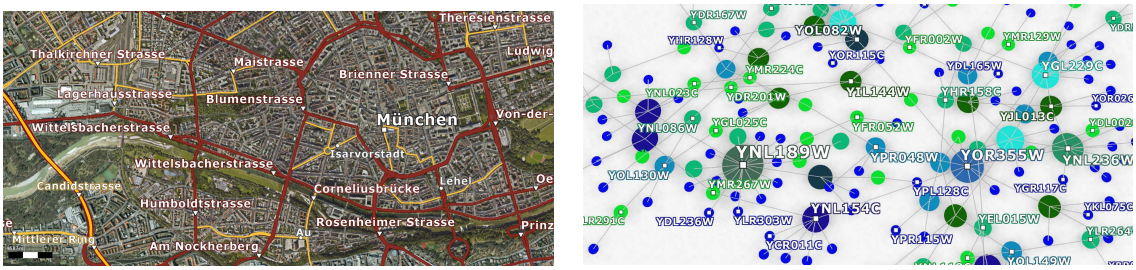


Figure 2.19: An example visualization of active range (AR) intervals for three labels proposed by Been *et al.* [13]. (a) The label size is represented in world coordinates, and as such, it is proportional to scale, meaning the screen size of the label is variant under zooming, but it remains the same relative to the view area. From a certain point, ARs begin to overlap with each other. (b) Therefore, active range optimization (ARO) is applied so that no two extrusions overlap and the sum of active range lengths is maximized. The black outline cone shows the view window. (c) Illustrates two horizontal slices at fixed scales $s = 1$ and $s = 2$ representing a 2D map. At $s = 1$, the gray label is outside of the view. However, as the user zooms out to scale $s = 2$, the gray label slides into the view and starts to overlap the other ARs. Therefore, the yellow label must disappear at some scale between $s = 1$ and $s = 2$.

2.3.1 Dynamic Internal Labeling

Been *et al.* [13] introduced a set of desiderata for dynamic internal labeling, which involves continuous zooming and panning. (1) Except for sliding in or out of the view area, labels should not vanish or appear when zooming in or out. (2) As long as a label is visible, its position and size should change continuously under the pan and zoom operations. (3) Except for sliding in or out of the view area, labels should not vanish or appear during panning. (4) The placement and selection of any label is a function of the current map state (scale and view area). The authors also proposed the first formal algorithmic framework where the placement (*i.e.*, size, orientation, and position) and selection (*i.e.*, labels that do not overlap with each other) of labels is made in the preprocessing stage, and only filtering (*i.e.*, labels that do not intersect the current view area are removed) is resolved during the interaction. The placement in a dynamic environment can be visualized as an extruded label shape along the vertical axis (scale); see Figure 2.19. Therefore, the dynamic label placement problem can be formulated as a static label placement extended by scale dimension. Label placement is done independently at each scale, abstracted from position models, allowing overlaps between labels during the preprocessing. However, although labels for each scale are placed separately, their positions relative to the associated feature must remain consistent across scales to ensure smooth transitions during zooming. The actual conflict resolution to avoid overlaps is handled in a subsequent process. A label can be selected only in the so-called *active range interval*, which is the range of scales during which a label is visible without overlapping other labels. Therefore, the objective is to determine the placement and range for each label such that no two extrusions overlap and the sum of active range lengths is maximized. In simpler terms, the goal is to maximize the visibility of as many labels as possible across various scales. The objective is known as *active range optimization* (ARO) problem, a dynamic counterpart to *label number maximization* (LNM) objective in static cases. Finally, they proved that ARO is $\mathcal{NP}$-hard and introduced a greedy technique for



(a) (b)

Figure 2.20: An example of label placement proposed by Vaaraniemi *et al.* [168]. (a) Label placement of a road network in Munich with a large buffer zone around labels to enhance readability and visual association. (b) Label placement of a protein interaction network with a small buffer zone that allows placing many labels at the cost of complicated visual association.

a simplified case running in polynomial time. Later, Been *et al.* [14] and Liao *et al.* [109] proposed several approximations of algorithms introduced by Been *et al.* [13].

Zhang *et al.* [187, 188] introduced a slightly different objective. Instead of maximizing the total visibility of labels (as in ARO), they focused on *maximizing minimum weighted active range* (MMWAR), which emphasizes fairness in visibility. In this formulation, labels are assigned weights based on their importance, and the goal is to ensure that even the least visible important label has a sufficiently long active range. Unlike ARO, which maximizing the overall number of visible labels, MMWAR ensures that critical labels (with higher weights) remain visible for an adequate amount of time.

While previous work [13, 14, 109, 187] is focused on zooming operations, Gemsa *et al.* [63, 64] introduced the ARO with map rotations, ensuring that the number of labels is maximized and labels remain horizontal during any rotation. They presented a constant-factor approximation for this problem based on line stabbing and refined it into an efficient polynomial-time approximation scheme (EPTAS). Later, Yokosuka and Imai [186] proposed an algorithm for label size maximization (LSM) on rotating maps while ensuring that the labels do not overlap with each other.

All previously mentioned methods take into account an entire map and optimize the label layout globally, regardless of the current view area. In contrast, Barth *et al.* [12], later extended by Gemsa *et al.* [62], developed a local view framework that optimizes the label placement for an animation of a map given offline as an input. Any label that is not visible in the current view area is ignored; thus, the problem size is also significantly reduced. The framework is abstracted from the particular map operations (*e.g.*, zooming, panning, rotation). The labeling problem is expressed as a set of time intervals representing the labels' presences, activities, and conflicts. A similar approach proposed Gemsa *et al.* [61], where a movement trajectory for the map is given as an input (*e.g.*, the path from point A to B in car navigation). Therefore, label placement can be optimized only along the trajectory. For the general problem, they presented an integer linear programming formulation and an approximation algorithm for a particular case of unit square labels.

Vaaraniemi *et al.* [168] introduced a force-based labeling algorithm for point, line, and area features of dynamic 2D and 3D scenes. The algorithm internally works entirely in the screen space – the 3D world coordinate of a feature is projected to 2D screen space. To resolve the temporal coherence, they apply several techniques. First, the optimal layout,

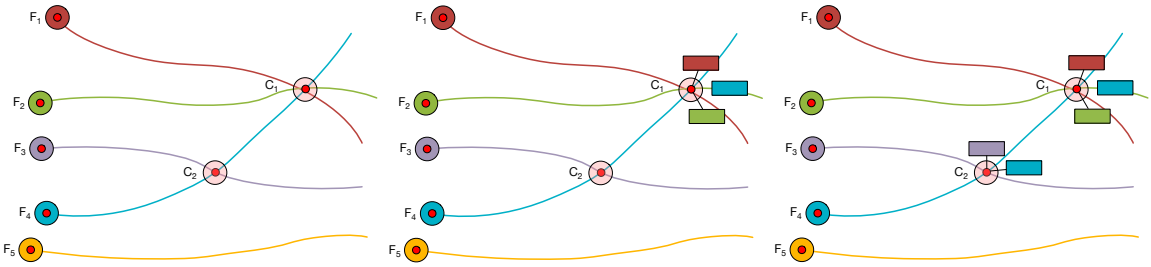


Figure 2.21: An example of spatiotemporal trajectories and sequential processing of intersections proposed by He *et al.* [73]. There are two joint sets (C1, C2). The introduced technique first performs static label placement on the frame where the joint sets are located, and the positions of labels are transferred to previous frames via constraints.

based on the previous one, is created for each frame. Second, the friction force is defined to allow only continuous changes. Third, the buffer zone is defined around each label where new labels can not be placed; see Figure 2.20. Finally, quickly moving labels and labels with acting forces that cancel each other out are removed (labels squeezed from multiple sides). Therefore, even if the feature is present and visible, it could be left unlabeled due to temporally coherent requirements or conflict of the corresponding label with the other one. The authors also conducted a preliminary study, where the participants (researchers or engineers in Human-Computer Interaction and visual designers) were asked several questions to define design principles. The one concerning point features was whether the labels should be scaled according to their depth (distance) in the 3D scene. The participants suggested that depth helps spatial perception in a 3D landscape. However, the scale factor should not exceed a certain minimum to maintain readability.

Bhore *et al.* [26] conducted an algorithmic study on fully dynamic independent sets for map labeling, explicitly addressing the challenge of point labeling with axis-parallel rectangular labels of uniform size, as well as uniform height and arbitrary width. Their objective was to maintain the largest possible set of non-overlapping labels in a dynamic environment where labels can be added or removed over time. This approach is relevant in scenarios where map features and labels appear and disappear over time, such as reports of earthquakes, forest fires, or disease incidences on social media. They introduced the first deterministic algorithm that maintains a maximal independent set (MIS) for a dynamic set of uniform rectangles. Additionally, they developed various deterministic dynamic approximation methods for solving the MIS problem.

He *et al.* [73] proposed a point-feature label placement technique using a slider model. The authors build on the method proposed by Vaaraniemi *et al.* [168] and extend it to the spatiotemporal trajectories of features. At the initial phase, the intersections of trajectories are placed by the force-based labeling method in the order given by the number of intersections; see Figure 2.21. The idea is to attract the position of corresponding labels in the previous frames to the positions at intersections. Finally, all forces are summed and solved by simulated annealing until the system reaches equilibrium, where the forces are balanced or a maximum number of iterations is reached.

In the context of dynamic internal label placement, the works of Petzold *et al.* [140] and Kouřil *et al.* [97], previously mentioned in Section 2.1, are also relevant here as they incorporate dynamic aspects into the label placement process. Petzold *et al.* [140] introduced a fast greedy slider model that addresses zooming and scrolling by splitting the process

into time-consuming preprocessing and a fast interaction phase. This method constructs a scale-independent conflict graph during preprocessing, which is later used for fast conflict resolution during interaction. Similarly, Kouřil *et al.* [97] extended internal label placement by developing a multi-scale, multi-level approach suitable for dense and multi-instance environments.

2.3.2 Dynamic External Labeling

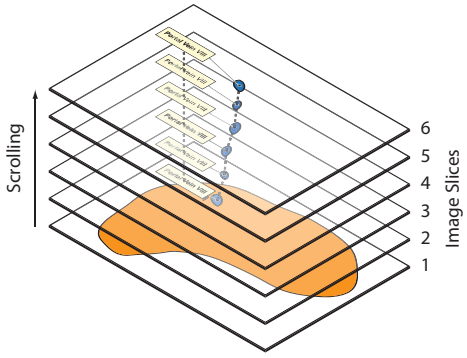
Preim *et al.* [141] proposed a method for temporally coherent boundary labeling focusing on 3D models for anatomy learning. The visualization is divided into a central part containing the visualized object and left and right parts used for textual descriptions. The proposed solution counts the number of anchors within the visible part of the object and adjusts the font size of textual descriptions based on this count. If the number of anchors exceeds a threshold, a conflict arises, necessitating a reduction in the number of anchors. The reduction leverages hierarchical relations among anchors, using group descriptions for sets of anchors within the same group. Although the proposed technique allows leaders to intersect, complicating the determination of correspondence between anchors and labels, it laid a solid groundwork and identified issues to be addressed in future research on external temporally stable label placement.

Azuma and Furmanski *et al.* [9] introduced a cluster-based approach where the overlapping labels are clustered. Within each cluster, the labels are repositioned simultaneously in the same position. For each cluster, 40-75 label positions are considered, including the same position as in the previous frame. The remaining sets of positions are chosen randomly. The best label position set is selected based on a cost function that penalizes label-label and label-feature overlaps and leader-line intersections or leader-label conflicts. However, the approach does not forbid these conditions to be accepted as a solution if the cost function value in frame $f_i \leq f_{i-1}$. The authors also provided a user study that compares users' response time of reading the label aloud of highlighted features among three additional techniques: greedy, simulated annealing (SA), and gradient descent. The results show that the users were the fastest for the SA technique even though it performed the most label transitions among the methods. However, the SA approach consistently achieved the best cost value, indicating the least number of conflicts in the label placement. Therefore, the authors claim that the conflicts influence the response time more than the temporal instability.

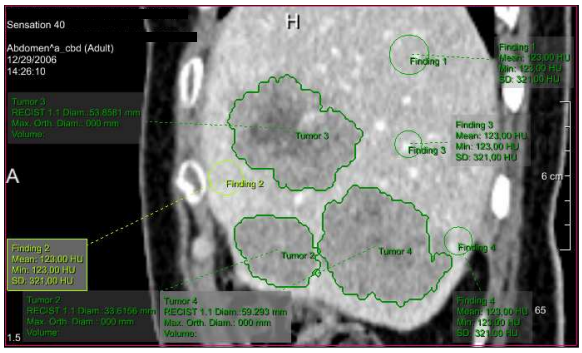
Ali *et al.* [4] proposed an anchor points stabilization based on an additional attractive force that aims to keep anchor points close to their previous positions. Čmolík and Bittner [40], and later Balata *et al.* [10] proposed a similar technique based on additional coherence terms for features and label positions.

Mühler and Preim [129] proposed a method for the labeling of 2D slices and 3D reconstructions of segmented medical structures for surgical planning. They propose to lock the once-calculated position of a label over multiple slices until the labeling is infeasible (*e.g.*, overlap of several labels); see Figure 2.22(a). A similar greedy approach was published by Mogalle *et al.* [127], who presented a more constrained label placement technique for labeling 2D slice data; see Figure 2.22(b).

Götzelmann *et al.* [65] proposed an algorithm to integrate internal and external labels and annotation boxes. Internal labels are in this work defined more as area labels, with anchors not precisely defined. The label candidates are tested sequentially based on predefined rules or simple strategies that aim to minimize overlaps and ensure readability,



(a) Slice coherency by Mühler and Preim [129]



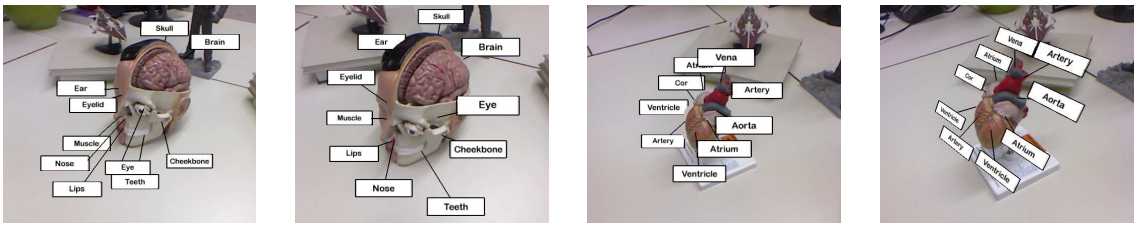
(b) Abdominal CT labels by Mogalle *et al.* [127]

Figure 2.22: (a) An illustration of slice coherency where the position of the label is fixed across multiple slices to maintain readability and tracking proposed by Mühler and Preim [129]. (b) Abdominal CT slice of the liver with labels placed by the technique of Mogalle *et al.* [127].

with the most relevant labels placed first. External labels are placed on the contour of an object, and overlaps among all types of labels are prohibited. The goal is to find a trade-off between static and dynamic labeling in a coherent manner. The authors propose using predefined local strategies to optimize the label layout and minimize displacement when interacting with an object. The first strategy involves slightly changing the position or size of the labels, and the second strategy is to place the label close to its location in the previous frame. Optimization is iterative, with agents continuously making minor adjustments until a satisfactory layout is achieved. The agents do not learn from experience but instead follow predefined heuristics and rules to optimize label placement. Therefore, the proposed approach lacks the adaptive learning capability of reinforcement learning agents.

Later, Götzelmann *et al.* [66] presented an approach focusing on the labeling of animated 3D objects such as combustion engines with moving pistons. The entire animation is analyzed to determine the calm and fluctuating regions. Afterward, the labels are placed in calm regions to move as little as possible during the animation. The label placement of fluctuating parts is resolved by visualizing the trajectory whose midpoint is connected to the corresponding label. Unfortunately, the approach considers the exact position and shape of illustrated objects and does not rely on any shape simplification with bounding objects.

Stein and Décoret [159] presented a greedy screen space approach for the dynamic labeling of interactive scenes based on energy optimization. The algorithm prioritizes features that are difficult to label, such as those within clusters or occupied regions. A shadow region is calculated for each feature, representing forbidden locations for labels, as leader lines would intersect already placed labels. This shadow area is subtracted from the available screen space, and each pixel within the remaining space is evaluated based on an objective function. The authors incorporated interpolation of label positions to maintain temporal coherence, bounding the velocity of moving labels to avoid visual discontinuity. This approach smooths label movements rather than fully solving the problem of temporal discontinuity. Later, Sirk *et al.* [157] proposed a similar greedy technique with a slightly modified energy function definition that better scales with a number of labels. Temporal coherency is maintained using forces moving labels from positions of the previous frame to-



(a) (b) (c) (d)

Figure 2.23: An example of label placement proposed by Tanzgern *et al.* [163] in various conditions. (a) Illustration of label stacking of center-based approach because poles have a similar orientation in 3D space. (b) Using plane-based strategy the issue can be resolved thought (c) it can suffer from label conflicts. (d) The plane-based approach can be relaxed such that a label is not always placed parallel to the view plane.

wards ideal positions in the current frame, similar to Hartmann *et al.* [69]. The results show that the labels can be placed closer to the features, and more labels are placed compared to Stein and Décoret [159].

Tanzgern *et al.* [163] proposed a 3D object space labeling approach (so-called *hedgehog* labeling). In contrast with most previously published methods, they define labels as elements of the 3D scene, which overcomes the lack of temporal coherence. Therefore, the leader is part of the line defined by the center of the sphere enclosing the illustrated object and the feature of the corresponding label in 3D space. The labels are always placed parallel to the view plane. They suggest two variants of the approach to prevent label occlusions: (1) *center-based method* is a 1DOF approach, where labels move along a 3D pole sticking out from the annotated object. However, this strategy tends to stack annotations if poles have a similar orientation in 3D space; see Figure 2.23(a). Therefore, this approach should be considered if anchor points are well distributed around the object of interest. (2) *Plane-base method* is a 3DOF approach, where the labels can move within a corresponding plane fixed in 3D space. Each label is assigned to the plane that is closest to its feature, and the planes are parallel to the view plane and are placed equidistantly; see Figure 2.23(b)-(c). Afterward, the labels are projected into the corresponding plane and placed using a force-based approach proposed by Ali *et al.* [4]. Finally, the placement is updated when the viewing angle to the plane is larger than a predefined threshold, which can lead to further difficulties resolved by relaxation of the label placement constraint (*i.e.*, a label is always placed parallel to the view plane); see Figure 2.23(d).

Maass and Döllner [117] proposed a similar hysteresis approach to make the movement of labels temporally coherent. During the interaction, the labels keep their current positions, and once the user pauses or finishes the interaction, the label layout is recalculated, and the labels perform a continuous movement to the computed position.

Madsen *et al.* [119] presented a study of temporal coherence strategies for augmented reality labeling. The authors analyzed three independent variables: (1) *labeling space* (2D image space or 3D object space), (2) *label layout update frequency* (discrete, continuous), (3) *feature point distribution* (balanced, unbalanced). During the trial, the duration and error rate was measured. The study results show that the discrete 3D object space method outperforms the continuous 2D image space labeling method (the authors suggest that the continuously updating layout makes it difficult for users to keep track of the labels) and even the discrete 2D image space labeling method. Nevertheless, the difference between

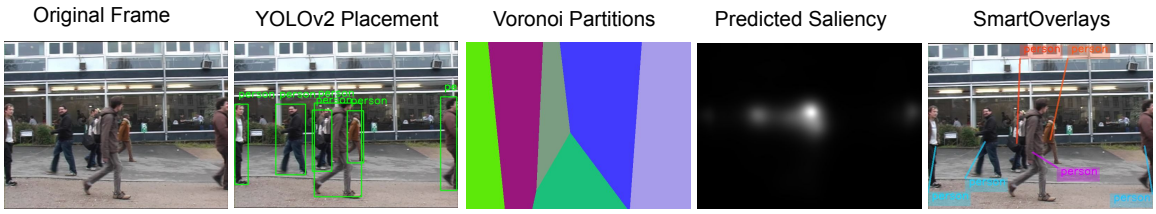


Figure 2.24: An illustration of input, leveraged modalities derived using machine learning (except Voronoi), and output of SmartOverlays proposed by Hegde *et al.* [74].

discrete and continuous 3D object space methods is not reported. Regarding the feature point distribution, they did not find any significant difference.

Hegde *et al.* [74] proposed SmartOverlays, a greedy framework for external label placement in augmented reality applications that addresses the common issues of clutter and occlusion. This method leverages a real-time object detector and a Saliency Attention Model (SAM) to generate bounding boxes and saliency maps for video frames. The label placement algorithm uses Voronoi partitioning, where the partitions are convex polygons, ensuring that leader lines are cross-free and labels are placed near their corresponding objects without overlapping or occluding critical visual elements; see Figure 2.24. The algorithm searches for the optimal label placement by finding the position with the minimum Label Occlusion over Saliency (LOS) score within each partition, determining the top left corner of the label. If multiple positions have the same minimum LOS score, the one closest to the centroid of the object's bounding box is selected. Temporal coherence is maintained by tracking objects across frames, reducing jitter in label placement. The technique also includes adaptive coloring for dynamic backgrounds, enhancing label readability, and ensuring clear association with objects of interest.

2.4 Labeling Guidelines

The effectiveness of label placement directly influences how easily and accurately users can interpret the information presented. Adequately placed labels not only enhance readability and reduce visual clutter but also contribute to the overall aesthetic quality of the visualization. Label placement guidelines are grounded in best practices that focus on factors such as legibility, unambiguity, and the prevention of conflicting labels. However, as is later discussed, there is not always a consensus on these principles, and practices vary across different bodies of work.

2.4.1 Internal Labeling Guidelines

The visual guidelines for label placement were established by Yoeli [185] and Imhof [82][6]. Their work, grounded in extensive practical experience, introduced a set of principles that have become widely recognized and adopted within the labeling community. The guidelines discussed herein are primarily based on the publications of Imhof [82] and Yoeli [185], which continue to influence contemporary practices in label placement across various fields.

[6]The original work *Die Anordnung der Namen in der Karte* was published in German in 1962.

G1. **Legibility.** Labels should be well-readable, easily discriminated, quickly, and easily located, even in densely populated areas. Legibility depends on factors such as font type, color, size, and the position or arrangement of other labels. The content of the visualization also plays a crucial role in determining legibility. Labels should not be broken into syllables. The ease of legibility and identification of the map's labels depends on (a) precise graphic relation between the label and the relevant item, (b) minimum of mutually disturbing interference between the labels and other contents of the map.

G2. **Unambiguity.** Each label should be associated with only one feature, avoiding any potential referential mismatches. Clear graphic association can often be determined by variations in style, size, size-gradation, and visualization content.

G3. **Conflictless.** Labels should not overlap with each other or with other visualization features, ensuring clarity and avoiding visual clutter.

G4. **Didactics/Semantics.** Labels should enhance and amplify the characteristics of the labeled objects, such as revealing spatial situations, territorial extents, connections, and importance. Didactical principles should be applied to reinforce the characteristics of the items being labeled (*e.g.*, , flowing placement of river labels, etc.).

G5. **Hiearchy.** Labels should reflect the hierarchical structure of the scene (*e.g.*, the population of the city). Labels of different sizes, font styles, or colors can be used to accomplish the hierarchical structure.

G6. **Aesthetics.** The dispersion of labels in the scene should avoid being too uniform or scattered. In order to accomplish this, less significant features can be removed. There should be free space around labels, and the map should not be overcrowded with names to maintain its optimal information value.

Imhof [82] explains that there are always exceptions to every rule, and often, rules may conflict with one another. Each specific case requires tailored consideration to determine the most appropriate principle to apply. In practice, adhering to all the rules relevant to a particular scenario is not always possible. Building on this notion, Wood [178] argues that cartographers should employ their overall judgment of these guidelines, selectively adhering to some while disregarding others, all with the objective of enhancing clarity. Imhof [82] also provides additional aesthetics and semantics recommendations specifically for point-feature label placement:

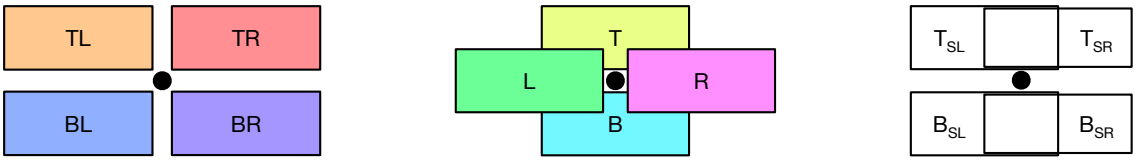
R1. Labels should be placed to the left of a river or boundary line for points on the left and to the right for points on the right. For locations straddling a river (*e.g.*, Budapest), the label should be split across the river or placed in available space to the right.

R2. For aesthetic reasons, labels should be entirely placed over land, avoiding placement over water bodies or partially between land and water.

R3. Labels for shoreline or coastal locations should be fully displayed over the water surface. Conversely, labels for places near but not directly on the shore should be positioned entirely on land.

R4. In dense, small-scale maps, all coastal labels should be positioned over the ocean. These labels should also be slightly curved outward from the horizontal line for better legibility.

R5. Narrow letter spacing should be chosen in small-scale maps to ensure clearer associations between labels and their corresponding locations



(a) (b) (c)

Figure 2.25: Common label positions for a point feature as presented in prior literature: (a) 4-position model, (b) additional positions of 8-position model, and (c) additional positions of 10-position model.

Only a few bodies of work elaborate further on the label placement principles. Formann and Wagner [54] claim that labels can usually be removed in geographical maps to avoid conflicts as small or less important cities can be left out. However, in technical maps, every feature is typically essential and needs to be labeled. The authors provide an example of groundwater maps in Munich, where the features are drill holes, and the label represents the name of the drill hole, the groundwater level, or similar relevant measurements. Van Dijk *et al.* [160] summarized the principles proposed by Imhof [82] and Yoeli [185] and proposed a conceptual definition of a quality function $q \in [0, 100]$, where $q = 0$ corresponds with the lowest quality and $q = 100$ with the highest quality. The function consists of aesthetics, label visibility, feature visibility, and label-feature association. The quality function also considers line and area features among point features. The authors provide an example definition of q, but as regards the point-features, the definition is relatively trivial. Aesthetics is set to a constant of 100, and the label's visibility is defined as the percentage of the area of a label that does not overlap with other elements, etc. Unfortunately, the authors have not provided any evaluation of the proposed technique with users or domain experts.

2.4.2 Candidate Position Guidelines

The primary challenge in internal point-feature label placement involves a selection of the most appropriate position for each label from among n potential candidates surrounding a point feature, typically with the objective of maximizing the number of labels that can be placed. When multiple suitable candidate positions are available for a point feature, the selection is determined based on the *Position Priority Order* (PPO) of the label candidates, an assigned value within the range $[1, n]$, where a lower value means a higher priority.

Within the cartographic community and the domain of automated label placement, a variety of position models have been utilized, differing in the number of label candidates considered for each point feature. The 8-position model is the most prevalent, evaluating eight potential label placements per point feature, followed in popularity by the 6-position, 4-position, and 10-position models. Figure 2.25 illustrates the label candidate positions for these models. Models with alternative configurations, such as the 5-position model, are used rarely. Alongside the position models, authors frequently delineate the priorities assigned to label candidates. However, the following review of the existing literature reveals inconsistencies and, at times, contradictions in these priorities. Table 2.4 shows the analyzed works detailing their chosen position models and the associated priorities.

As already mentioned, the initial guidelines for selecting appropriate label candidates, among other labeling principles, were formulated by Imhof in 1962, published in German, and later translated into English [82]. Imhof introduced a 5-position model tailored for

Author	Year	Citations	Type	Position Priority Order									
				1	2	3	4	5	6	7	8	9	10
YoeliA [185]	1972	231	A	TR	TL	BR	BL	T	B				
Robinson *et al.* [146]	1995 (6th)	3096 (4th)	B	TR	TL	BR	BL	T	B	T_{SR}	B_{SR}	T_{SL}	B_{SL}
Brewer [33]	2015	441	B	TR	TL	BR	BL	T	B				
YoeliB [185]	1972	231	A	TR	TL	BR	BL	R	L	T	B		
Dent [47]	2009 (6th)	1304 (5th)	B	TR	TL	BR	BL	R	L	T_{SL}	B_{SR}		
Dobias (QGIS)	2009	–	SW	TR	TL	BR	BL	R	L	T_{SR}	B_{SR}		
Krygier *et al.* [98]	2016 (3rd)	361	B	TR	TL	BR	BL	R	L	T_{SL}	B_{SR}		
Christensen and Marks [39]	1995	520	A	TR	TL	BL	BR	R	T	L	B		
Yamamoto [182]	2005	40	A	TR	TL	BL	BR						
Ebinger and Goulette [50]	1989	–	A	TR	BR	TL	BL						
Wood [178]	2000	28	A	TR	BR	TL	BL	T_{SR}	B_{SL}				
Slocum *et al.* [158]	2022 (4th)	962	B	TR	BR	TL	BL	T	B	R	L		
Imhof [82]	1975 (*1962)	439 (*88)	A	TR	R	T	B	L					
Zoraster [191]	1986	108	A	TR	T	R	TL	BR	L	B	BL		
Jones [86]	1989	79	A	TR	R	BR	TL	L	BL				
Zoraster [192]	1990	129	A	TR	T	TL	R	L	BR	B	BL		
Zoraster [193]	1997	103	A	T	TR	TL	R	L	BR	B	BL		
`PerceptPPO`	2024	–	A	T	B	R	TR	BR	L	TL	BL		

Table 2.4: This overview outlines the PPO recommendations of various authors grouped by similarity. The literature is categorized into four groups based on the similarity of their priority schemes. A fifth category encompasses works with distinct priority practices that do not align with those of any other group. The top-right (TR) positioning emerges as a dominant preference, with authors consistently favoring it. While TR remains widely accepted across the board, there is noticeable diversity in selecting other positions. The citation counts are derived from Google Scholar. *The seminal work, *Die Anordnung der Namen in der Karte*, was first introduced in German in 1962. Yoeli [185] and Zoraster [193] define more than eight positions not included in our comparison as they are rarely used. Abbreviations: A – article, B – book, SW – software.

left-to-right languages, leveraging his cartographic expertise to recommend the top-right position as the most favorable for label placement. His preference was rooted in typographic principles, *e.g.*, the top position (T) was favored over the bottom (B). This rationale was based on the observation that in the Latin alphabet, ascenders are more common than descenders, suggesting that labels placed at the top are likely to appear visually closer to their corresponding point features. This consideration is particularly relevant for city names, which typically begin with a capital letter.

Yoeli [185] introduced the first algorithms for the automated positioning of point-feature labels. He proposes two n-position models later adopted for point-feature label placement by other authors. The first position model, denoted in Table 2.4 as *YoeliA*, is a 10-position model where the label candidates are organized around the point feature as in Figure 2.25. Notably, Yoeli proposed a grid system for typesetting the labels where the size of the grid cell is based on the size of the letters. Therefore, Yoeli distinguishes between labels with an odd and even number of letters. Since the latter cannot be centered above or below the point feature, Yoeli introduces additional top and bottom position modifiers that shift the label slightly left (SL) or slightly right (SR). In Table 2.4 are reported only the first six positions, as nowadays, even labels with an even number of letters can be easily centered

above or below the point feature. Similarly, Brewer [33] omits the four additional positions with shifted labels in his 6-position model. For Robinson *et al.* [146], Table 2.4 reports all positions in their 10-position model with priorities identical to Yoeli's, as they do not provide any information as to why the positions with shifted labels are included in his model.

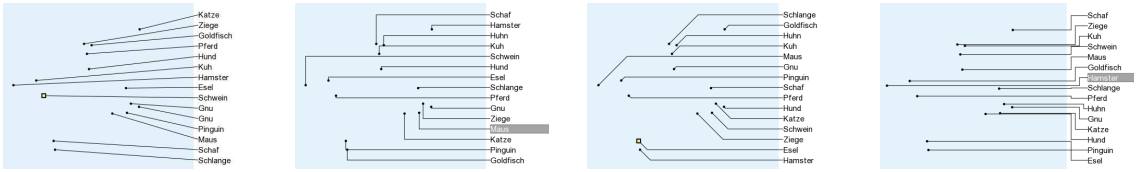
The second position model, denoted in Table 2.4 as *YoeliB*, is an 8-position model that Yoeli originally crafted for labeling small-area features. Nonetheless, other authors [47, 98] adopted or adapted Yoeli's 8-position model for point-feature label placement. Similarly, as in the previous 10-position model, Yoeli [185] uses additional top and bottom positions slightly shifted to the left for labels with an even number of letters. However, the overview presented in Table 2.4 again ignores additional adjustments for even-lettered labels, as they can be precisely centered directly above or below the point feature nowadays, regardless of letter count.

Wu and Buttenfield [180], although considered controversial by Mills [124], questioned the PPO proposed by examining road maps produced by three separate publishers, to determine if Yoeli's prioritization exists and is consistent with that on medium-scale navigational maps, and the extent to which map producers vary in their prioritization of positioning point-feature labels. They found that label candidate positions varied significantly between publishers. They also identified that the top (T) and bottom (B) positions were more commonly used than Yoeli's model suggested. Finally, the data did not support Yoeli's prioritization, indicating that it might not align with practical cartographic applications and is influenced by individual publisher preferences. The authors concluded that while Yoeli's model has been influential, its practical application varies, and alternative prioritization strategies may better reflect cartographic practices.

Christensen and Marks [39] introduced two algorithms for PFLP, employing gradient descent and simulated annealing techniques. The formulation of the proposed objective function draws upon Yoeli's foundational work [185]. Notably, they adopt an 8-position model similar to Yoeli's, but with swapped priorities of bottom-left (BL) and bottom-right (BR) positions and the top (T) and left (L) positions, which they describe as a standard PPO.

Ebinger and Goulette [50] proposed a 4-position model, see Figure 2.25(a), which diverges in priority schemes from one proposed by Imhof [82] and Yoeli [185]. While Yoeli prioritizes top positions over bottom positions, Ebinger and Goulette prioritize positions on the right side over the left. Wood [178] proposed a 6-position model with the top position being shifted slightly right (T_{SR}) and the bottom slightly left (B_{SL}). The author remarks that the shifted positions should only be used in extreme cases. Moreover, he argues that the shifted positions help associate the label with the feature. Unfortunately, no justification is given for this claim. Similarly, Slocum *et al.* [158] proposed an 8-position model extending Wood's first for positions to include top (T), bottom (B), right (R), and left (L) positions, enhancing label placement flexibility.

The prior literature exhibits an even more significant variety in priority schemes for label placement. Zoraster [191–193], in his series of works, introduced three distinct 8-position models for oil well labeling, each with unique priorities. A notable trend across Zoraster's models is the elevated priority given to the top position, starkly contrasting with other authors' approaches. Additionally, Jones [86] proposed 8-position models that resemble the 4-position model by Ebinger and Goulette [50], with a nuanced approach to prioritization: the right (R) position is placed between the top-right (TR) and bottom-right



(a) `s`-leader (b) `po`-leader (c) `do`-leader (d) `opo`-leader

Figure 2.26: Example of stimuli for leader type readability study conducted by Barth *et al.* [11]. Participants were asked to select label or feature corresponding to a highlighted feature or label.

(BR) positions, while the left (L) position's priority is set between the top-left (TL) and bottom-left (BL) positions.

Upon examining Table 2.4, it becomes clear that except Zoraster [193], the top-right (TR) position emerges as the highest priority across multiple bodies of work. The designation of priorities to other positions shows even more significant variability, highlighting a lack of consensus. Moreover, all reported priorities are based on the experience of the authors, and none of them were empirically verified by users. To our knowledge, the sole exception is a study by Scheuerman *et al.* [151], which attempts to evaluate position priorities with user input but limits its scope to just three positions (TL, L, BL). The study establishes the position preference order of $L > TL > BL$, indicating L is favored over TL and BL, and TL is preferred over BL.

2.4.3 External Labeling Guidelines

Regarding the external label placement, Barth *et al.* [11] conducted a user study, where they evaluated boundary labeling with different leader types (`s`-, `po`-, `do`-, `opo`-leaders) with respect to their performance, *i.e.*, whether and how fast a viewer can assign a feature to its label and vice versa; see Figure 2.26. They found that `do`-leaders perform best in the preference rankings, but concerning the assignment tasks, they perform slightly worse than `po`- and `s`-leaders. Nevertheless, concerning other factors, they recommend `po`-leaders as the best compromise between measured task performance and subjective preference.

Hartmann *et al.* [70] analyzed hand-drawn anatomical illustrations and created several metrics for functional requirements and aesthetic attributes of internal and external label placement. *Readability* is affected by label placement and font attributes. Therefore, labels and leaders should not overlap with each other or with visualized objects. The text should be oriented horizontally, and minimal contrast between lettering and its neighborhood should be met. *Unambiguity* guarantees that the association between features and labels prevents referential mismatches. Labels should be placed as close as possible to their corresponding features. The number of leader bends should be minimized. Features should not form clusters. *Aesthetics* should be maintained by symmetric layouts and visual clutter should be avoided. If an internal label cannot be placed aesthetically, it should be placed externally instead. The distributions of features and labels should not be too scattered or uniform. External labels should be aligned with each other or along the silhouette of a visualized object. *Temporal Coherency* should be maintained to prevent visual discontinuities during user interactions. The distance of identical elements between subsequent frames should be minimized.

Later, Niedermann *et al.* [133] conducted interviews with domain experts, and based on a semi-automatic analysis of 202 handmade anatomical drawings, they identified a set of 18 layout quality criteria. These include labels placed outside of the drawing area, no conflicts among labels, uniform spacing between labels, and the same font for all labels. Labels should consist of single-line text, the length of leaders should be minimized, and leaders running parallel should be avoided as it can cause a referential mismatch. Furthermore, they designed a new geometric label placement algorithm based only on the most important criteria. They formally proved that the approach yields labeling that satisfies all hard constraints and has a minimum overall cost. Moreover, they showed on real-world anatomical drawings that the resulting labeling is of high quality and can be produced in adequate time.

Further labeling knowledge can be transferred from the general drawing principles such as maximizing symmetry, minimizing edge crosses, and minimizing bends [142, 176].

Chapter 3

Temporally Stable Optimization Approach to Boundary Labeling

Interactive applications of labeling algorithms introduce a new aspect of temporal coherence. Applying only static algorithms on a frame-by-frame basis leads to temporally unstable behavior. In such a case, labels often jump abruptly from one position to another, breaking several assumptions of high-quality labeling. In this chapter, we specifically focus on the one-sided boundary labeling of dynamic scenes, where labels are placed on the top of the scene (the static case was introduced by Gemsa *et al.* [60] as *panorama* labeling). The features in the scene are approximated by points denoted as *anchors*. The visual relationship between labels and corresponding features is established by vertical leaders that connect the label with the anchors; see Figure 2.12(c).

We propose two labeling methods suitable for a diverse range of applications. The first is designated for the offline processing of the entire interaction in advance. Such a method can be valuable for creating, *e.g.*, educational visualizations, television news infographics, or generally in the movie industry, where the complete interaction with the scene (*i.e.*, all the frames of the video-sequence) is known in advance. Imagine video-footage from a drone flying through mountain terrain or a city, where one would like to label peaks or tourist attractions, respectively. The second method is designed for the online processing of continuously delivered frames created on-demand as a result of interacting with a dynamic scene. Such a method can be applied in, *e.g.*, games, 3D map viewers, and augmented or virtual reality applications. Imagine an interactive application presenting a 3D map (digital elevation model), where one would like to know nearby points of interest and could move along the scene by interacting with the camera (*e.g.*, pan or rotate).

We claim the following contributions: (**1**) A temporally stable labeling method designed for the offline processing of the entire interaction with the scene in advance. (**2**) A novel temporally stable labeling method designed for interactive visualizations. (**3**) An extended labeling terminology of Bekos *et al.* [20] suitable for interactive and non-interactive labeling of dynamic scenes. (**4**) A formulation of visibility optimization based on feature prominence, and an extension for smooth label transitions. (**5**) A comparison of the proposed methods with three others, and the results of an extensive user study on several aspects of labeling.

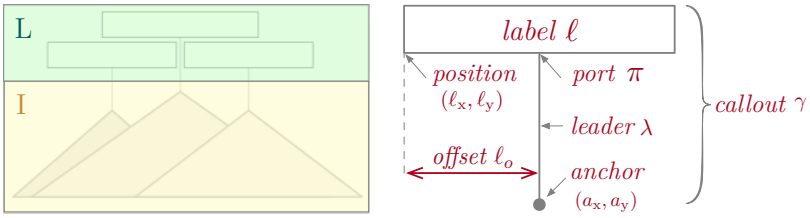


Figure 3.1: Illustration of terminology based on the report of Bekos *et al.* [20].

3.1 Problem Definition and Terminology

In this section, we describe the formal definition of boundary labeling for dynamic scenes. We extended the terminology and definitions proposed by Bekos *et al.* [20], that are suitable for *static* labeling, to capture specific aspects of *dynamic* labeling. As input we are given a sequence S of frames $F_0, F_1, \ldots, F_{n=|S|}$. A frame is a tuple $F = (D, \mathcal{A}, \mathcal{L})$ described by a *drawing region* D partitioned into the *image region* I and *labeling region* $L = D \setminus I$; see Figure 3.1. The drawing region D has the same dimensions $(D_\mathrm{w}, D_\mathrm{h})$ for the entire sequence S. A set of *anchors* $\mathcal{A}$ denotes the points of interest to be labeled. Each anchor a is a point of the image region I with coordinates $(a_\mathrm{x}, a_\mathrm{y})$. Furthermore, each anchor has additional information attached (*e.g.*, the name of the anchor). The axis-aligned bounding box of additional information is denoted as *label* ℓ of dimensions $(\ell_\mathrm{w}, \ell_\mathrm{h})$. Each label is a rectangular sub-region of L placed at coordinates $(\ell_\mathrm{x}, \ell_\mathrm{y})$; see Figure 3.1. We denote a set of all instances (*e.g.*, labels, anchors) in frame F with a superscript such as $\mathcal{L}^F$ (labels) and a set of all instance that occur at least once in any frame of sequence S such as $\mathcal{L}^S$. Each anchor is connected to its label ℓ by a *leader* λ at an attachment point called *port* π on the boundary of ℓ. The distance between ℓ_x and π is called *offset* ℓ_o (*i.e.*, horizontal coordinate $\ell_\mathrm{x} = a_\mathrm{x} - \ell_\mathrm{o}$). A *callout* is the collection $\gamma = (\lambda, \pi, \ell)$ of a leader λ connected to a label ℓ at the point π. A set $\mathcal{C}$ of callouts is called *labeling* or *label layout*. A labeling of frame F and sequence S is denoted as $\mathcal{C}^F$ and $\mathcal{C}^\mathcal{S}$, respectively. We call a labeling $\mathcal{C}$ of a sequence S *valid* if it satisfies the following requirements.

(**R1**) The labels do not overlap with each other [4, 13, 70, 117, 129].

(**R2**) The labels are connected with the corresponding anchors with vertical leaders [60, 70, 117, 129].

The *labeling quality* among a set $\mathcal{S}$ of all valid labelings can be described by a cost function $c : \mathcal{S} \to \mathbb{R}^+$. The *optimal labeling* $\mathcal{C}^0 \in \mathcal{S}$ satisfies the condition of optimality $c(\mathcal{C}^0) < c(\mathcal{C}) \forall C \in \mathcal{S}$ and respects all the following criteria C.

(**C1**) The number of stacked layers of labels in the label layout is minimized [4]. To put it differently, the labels are placed as close as possible to the corresponding anchor [70, 117, 129].

(**C2**) The leader is connected to the label as close to the center of the label as possible to provide clear mapping [117], and to achieve aesthetic and symmetric layout [60, 70].

(**C3**) The movement of the labels through the interaction is temporally coherent [4, 129]. In other words, the vertical and horizontal movement of the labels should be continuous without abrupt changes [13], and minimized through the interaction with the scene [70].

(**C4**) [Optional] The vertical positions of labels should correspond to the distances of labeled anchors in a scene from the camera center. The labels of the closest anchor should be the lowest in the label layout [117].

Please note, that high-quality human-friendly labeling, in general, is hard to formalize as it is relative to various subjective and domain aspects. Therefore, high-quality boundary labeling is often a compromise among the described criteria. Because of the previous statement, we provide a quantitative evaluation in Section 3.6.1 and an extensive user study in Section 3.6.2–3.6.3.

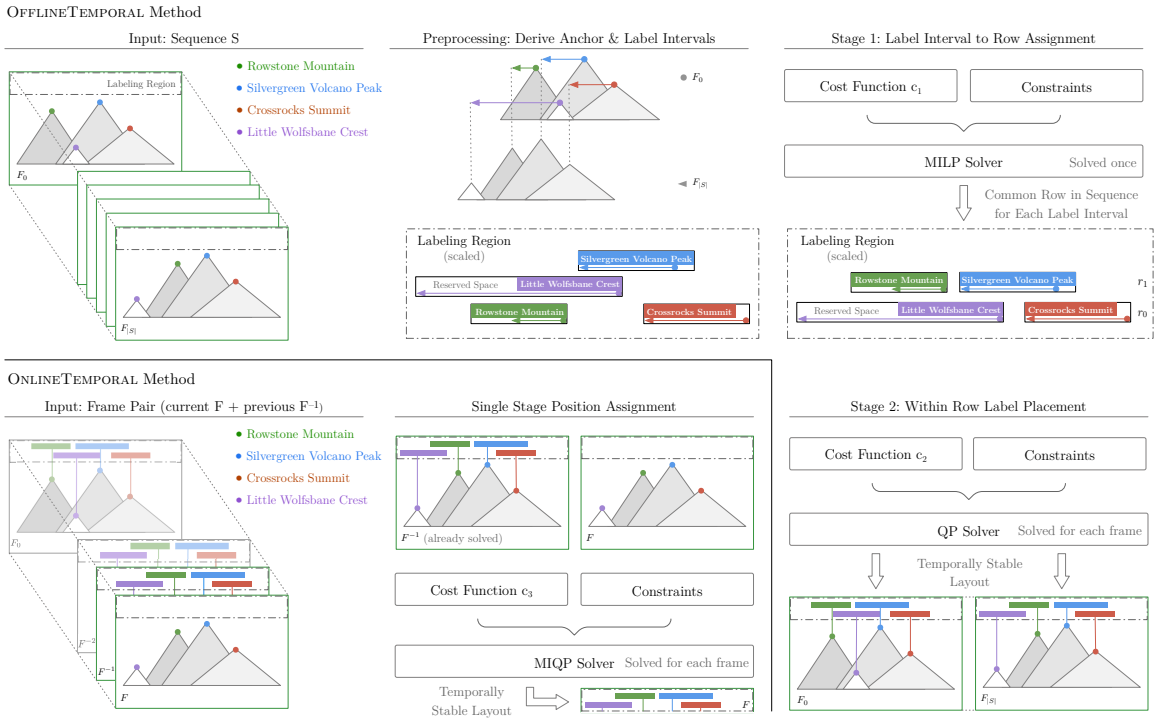
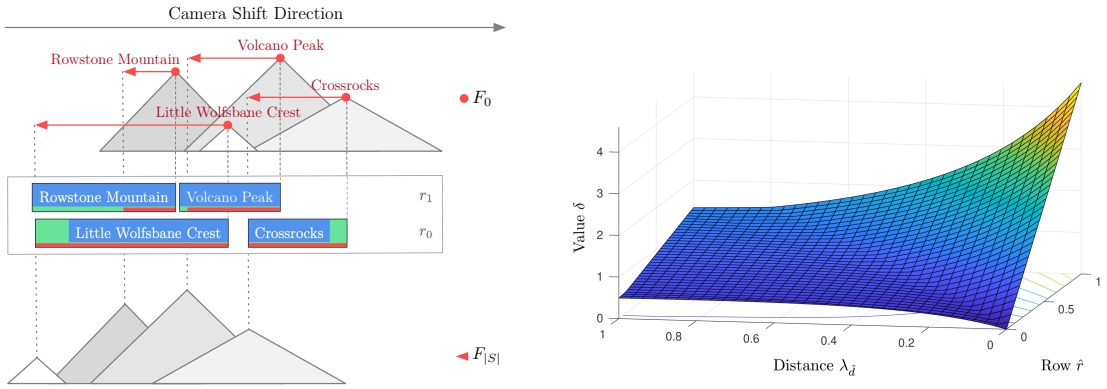


Figure 3.2: High-level overview of processing stages applied in the proposed methods. The OfflineTemporal method consists of the *preprocessing* stage followed by the *Label Interval to Row Assignment* and *Within Row Label Placement* stage. The OnlineTemporal is designed as a single-stage method.

3.2 Offline Optimization

When the entire interaction with the scene is known in advance, we process the sequence S as a whole. We place labels into discrete rows such that the vertical position ℓ_y is approximated by *row* $r \in R$, where $R = \{\, r \mid r \in \mathbb{N} \;\wedge\; r \leq |\mathcal{A}^S| \,\}$ is a set of available rows. Furthermore, we suppose the width ℓ_w and the height ℓ_h is fixed for all $F \in S$. The proposed temporally coherent method denoted as OfflineTemporal consists of the following three stages (for a high-level overview, see Figure 3.2).

Preprocessing. Given a set of anchors $\mathcal{A}^S$, we create an *anchor interval* $\alpha = [\alpha_\mathrm{min}, \alpha_\mathrm{max}]$ for each anchor $a \in \mathcal{A}^S$. The anchor interval α captures the horizontal movement of anchor a through all the frames $F \in S$. The minimum x-coordinate $\alpha_\mathrm{min} = \min(a_x)$ and the maximum $\alpha_\mathrm{max} = \max(a_x)$ of an anchor define the bounds of the corresponding anchor interval α; see Figure 3.3(a). We denote the set of anchor intervals derived from the sequence S by A^S.



(a) (b)

Figure 3.3: (a) Example of sequence preprocessing in the OfflineTemporal method. The movement of the anchor in the x-axis starts at the position denoted by a red circle in frame F_0 and ends at the position denoted by a red triangle in frame $F_{|S|}$. The label interval λ (a green rectangle with a black stroke) reserves the space for the horizontal movement of its label. The length of the label interval λ is derived from the label width ℓ_w (blue rectangle) and the length of anchor interval α (red line). The left bound of label interval λ_min and its vertical position (row r) is optimized in *Label Interval to Row Assignment* stage. (b) Function δ with parameters $w_1 = 0.1$, $w_2 = 0.8$ and $w_3 = 0.5$.

Given a set $\mathcal{L}^S$ of labels, we then define a *label interval* $\lambda = [\lambda_\mathrm{min}, \lambda_\mathrm{max}]$ for each label $\ell \in \mathcal{L}^S$. We construct the label interval λ to reserve space for the horizontal movement of the label. This way, we can fixate the vertical movement of a label and allow movement only in the horizontal direction. Therefore, one label cannot influence the movement of any other label, which fulfills Criterion 3. We derive the width of a label interval λ from the associated anchor interval a and the label ℓ as $\lambda_\mathrm{w} = \max(\alpha_\mathrm{max} - \alpha_\mathrm{min}, \ell_\mathrm{w})$. We denote the set of label intervals derived from the sequence S by Λ^S. Furthermore, we calculate an average camera-to-anchor distance α_d for each anchor $a \in \mathcal{A}^S$ to be able to satisfy Criterion 4. Because the label interval λ is associated with the anchor interval α, the camera-to-anchor distance $\lambda_\mathrm{d} = \alpha_\mathrm{d}$.

Label Interval to Row Assignment. In this stage, we determine the vertical position $\ell_\mathrm{y} = r$ and the left bound λ_min of the label interval λ. The right bound of the label interval λ can then be derived as $\lambda_\mathrm{max} = \lambda_\mathrm{min} + \lambda_\mathrm{w}$. Since the label moves inside the reserved space given by its label interval λ, the label intervals must not overlap with each other (Requirement 1). Similarly, for the same reason, the anchor interval α must be a sub-interval of label interval λ; otherwise, Requirement 2 could be violated (label ℓ could not be connected with its anchor a). Furthermore, as we determine the row r in which the label will be fixed throughout the sequence S, we also take Criterion 1 and Criterion 4 into account in this stage. Please note that this stage is solved only once for the given sequence S. For an example of a label-to-row assignment; see Figure 3.3(a).

Within Row Label Placement. In this stage, we derive the horizontal position ℓ_x of label ℓ fixed within row r by optimizing the offset ℓ_o from the port π for each frame $F \in S$. Therefore, we take into account Criterion 2. However, in order to better fulfill Criterion

2 at the bounds of λ, we enable the label to pop out of its associated label interval λ. Consequently, we consider also Requirement 1 and Requirement 2.

3.2.1 Label Interval to Row Assignment

We formulate the problem as mixed-integer linear programming (MILP), which combines combinatorial optimization over binary variables with linear optimization over continuous variables [22]. The instance of MILP is formulated as the minimization of the cost function c_1 with respect to decision variable $I_\lambda^r \in \{0,1\}$ that indicates whether the label interval λ is placed in row r, and with respect to continuous variable λ_{min} considered further in the definition of constraints. The cost function is defined as

$$c_1\left(\Lambda^S\right) = \sum_{\lambda \in \Lambda^S} \sum_{r \in R} I_\lambda^r \hat{r} + I_\lambda^r \delta(\lambda_{\hat{\text{d}}}, \hat{r}). \tag{3.1}$$

The hat modifier in the above-given variable (*e.g.*, $\hat{r}$) denotes the unity-based normalized value of that variable. The product in the first term of c_1 supports Criterion 1. The function $\delta(\hat{d}, \hat{r})$ in the second term of c_1 is defined as

$$\delta\left(\hat{d}, \hat{r}\right) = \frac{\left|\hat{r} - w_1 \hat{d}\right| + \left|w_2 \hat{r} - \hat{d}\right| + w_3 \left|\hat{r} - \hat{d}\right|}{\left(\hat{d} + w_2\right)^2} \tag{3.2}$$

and supports Criterion 4. The purpose of the δ function is to establish a relation between normalized distance $\hat{d}$ and the row r of the label interval λ. By observing the influence of various values of the weights w_1, w_2 and w_3 on the resulting layouts, we recommend using the weights $w_1 = 0.1$, $w_2 = 0.8$, and $w_3 = 0.5$; see Figure 3.3(b). Please note that we have selected these values with Criterion 4 in mind.

To fulfill Requirement 1 and Requirement 2, we define the following constraints. First, we define the constraint to satisfy Requirement 1 as

$$\lambda_{\text{min}}^{(i)} + \lambda_{\text{w}}^{(i)} \leq \lambda_{\text{min}}^{(j)} + M \cdot \left(1 - I_{\lambda^{(i)}}^r\right) + M \cdot \left(1 - I_{\lambda^{(j)}}^r\right), \tag{3.3}$$

where we define the order so that the associated anchor interval $\alpha_{\text{min}}^{(i)} \leq \alpha_{\text{min}}^{(j)} \wedge \alpha^{(i)} \neq \alpha^{(j)}$ and $\alpha^{(i)}, \alpha^{(j)} \in A^S$. This constraint only needs to be applied in the case that both label intervals are in the same row r which is indicated by the binary decision variables $I_{\lambda^{(i)}}^r$ and $I_{\lambda^{(j)}}^r$. The use of a binary variable to activate and deactivate the constraint is a known trick in MILP [37, 60]. The constant M needs to be sufficiently large to deactivate the constraint (*i.e.*, the constraint is always true for any combination of $\lambda^{(i)}$ and $\lambda^{(j)}$ that are not in the same row). We set M equal to the width of drawing region D_w.

From the definition of the label interval λ and from Requirement 2 it follows that the interval must completely overlap its associated anchor interval α. Therefore, we introduce constraints to enforce that α is the subinterval of λ as

$$\lambda_{\text{min}} \leq \alpha_{\text{min}}, \tag{3.4a}$$

$$\lambda_{\text{min}} + \lambda_{\text{w}} \geq \alpha_{\text{max}}. \tag{3.4b}$$

Finally, the label interval λ is allowed to occupy only a single row r. Therefore, we define this restriction as

$$\sum_{r \in R} I_\lambda^r = 1. \tag{3.5}$$

3.2.2 Within Row Label Placement

We formulate the problem as convex quadratic programming (QP). When each label interval is assigned to a row and its left bound λ_{min} is set, it remains to determine the vertical position of each label for a given frame F so that Criterion 2 is reflected. The instance of QP is formulated as the minimization of the cost function c_2 with respect to the continuous offset variable ℓ_{o}. The cost function for the given frame F is defined as

$$c_2(F) = \sum_{\ell \in \mathcal{L}^F} \left(\ell_{\text{o}} - \frac{\ell_{\text{w}}}{2} \right)^2. \tag{3.6}$$

The function c_2 enforces Criterion 2 only. To enforce Requirement 1, we define a constraint for each pair of labels $\ell^{(i)}$ and $\ell^{(j)}$ associated with anchors $a_{\text{x}}(\ell^{(i)})$ and $a_{\text{x}}(\ell^{(j)})$ in the given frame F as

$$a_{\text{x}}\left(\ell^{(i)}\right) - \ell_{\text{o}}^{(i)} + \ell_{\text{w}}^{(i)} \leq a_{\text{x}}\left(\ell^{(j)}\right) - \ell_{\text{o}}^{(j)}, \tag{3.7}$$

where we suppose an order so that $a_{\text{x}}(\ell^{(i)}) < a_{\text{x}}(\ell^{(j)}) \wedge \ell^{(i)} \neq \ell^{(j)}$ and $\ell^{(i)}, \ell^{(j)} \in \mathcal{L}^F$. Furthermore, to satisfy Requirement 2 we define the constraints

$$\ell_{\text{o}} \geq 0, \tag{3.8a}$$
$$\ell_{\text{o}} \leq \ell_{\text{w}}. \tag{3.8b}$$

Finally, we want to restrict a label overflow with vertical bounds of drawing region D_w. This is accomplished by a pair of

$$a_{\text{x}}(\ell) - \ell_{\text{o}} \geq 0, \tag{3.9a}$$
$$a_{\text{x}}(\ell) - \ell_{\text{o}} + \ell_{\text{w}} \leq D_w. \tag{3.9b}$$

3.3 Online Optimization

The previously described method in Section 3.2 is not suitable when the entire interaction with the scene is not known in advance. Continuous delivery of frames makes it impossible to retrieve the anchor and label intervals. Furthermore, the performance of the method is also an aspect of concern in interactive applications.

Therefore, we propose an interactive method OnlineTemporal that removes described pitfalls and at the same time reflects Requirement 1 and Requirement 2 while satisfies Criterion 1–Criterion 4. The OnlineTemporal method, in contrast to OfflineTemporal, processes the entire interaction frame by frame and consists of only single stage wherein the position of the label is determined at once individually for each F (see a high-level overview in Figure 3.2). We again place labels into discrete rows such that the vertical position ℓ_y is approximated by row $r \in R$, where $R = \{\ r \mid r \in \mathbb{N} \ \wedge \ r \leq |\mathcal{A}^F|\ \}$. Furthermore, we suppose that the width ℓ_{w} and the height ℓ_{h} are consistent for each F.

Single Stage Position Assignment. Given a frame F and its immediate predecessor F^{-1}, we determine the temporally stable position $(\ell_{\text{x}}, \ell_{\text{y}})$ for each label ℓ at once and without any knowledge of the following frames. Unlike the OfflineTemporal method, we can not fixate the vertical movement of a label due to its uncertain unfolding in the future.

Therefore, we allow a label to change its vertical position; nevertheless, in the definition of the cost function, we minimize this behaviour in favor of Criterion 3. Furthermore, we restrict the vertical change of a label within two consecutive frames to be at most a single row (*i.e.*, row ℓ_{r}^{F} of label ℓ in frame F is either $\ell_{\mathrm{r}}^{F} = \ell_{\mathrm{r}}^{F^{-1}}$ or $\ell_{\mathrm{r}}^{F} = \ell_{\mathrm{r}}^{F^{-1}} \pm 1$). This technique also narrows down the optimization search space, which in turn speeds up the computation of subsequent frames.

3.3.1 Single Stage Position Assignment

We formulate the problem as mixed-integer quadratic programming (MIQP). The instance of MIQP is formulated as the minimization of the cost function c_3 with respect to decision variable $I_\ell^r \in \{0, 1\}$, that indicates whether a label ℓ is placed in row r, and with respect to continuous offset variable ℓ_{o}. The cost function for any given frame F is defined as (for simplification the superscript of a current frame is omitted)

$$c_3\left(F^{-1}, F\right) = \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I_\ell^r \left(w_{row}\hat{r} + w_{dist}\delta(\ell_{\hat{\mathrm{d}}}, \hat{r})\right) \tag{3.10a}$$

$$+ \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I_\ell^r \left(w_{row_\Delta}\left(r - \ell_{\mathrm{r}}^{F^{-1}}\right)^2\right) \tag{3.10b}$$

$$+ w_{offset} \sum_{\ell \in \mathcal{L}^F} \left(\frac{1}{\ell_{\mathrm{w}}}\left(\ell_{\mathrm{o}} - \ell_{\mathrm{o}}^{F^{-1}}\right)\right)^2 \tag{3.10c}$$

$$+ w_{center} \sum_{\ell \in \mathcal{L}^F} \left(\frac{1}{\ell_{\mathrm{w}}}\left(\ell_{\mathrm{o}} - \frac{\ell_{\mathrm{w}}}{2}\right)\right)^2. \tag{3.10d}$$

The variable ℓ_{d} denotes the distance of anchor $a_{\mathrm{x}}(\ell)$ associated with label ℓ from the camera center. Further definitions of the delta function $\delta(\hat{d}, \hat{r})$ and hat modifier from Section 3.2.1 hold. The first term (3.10a) reflects the definition of a cost function c_1 from Section 3.2.1; therefore supports Criterion 1 and Criterion 4. The second term (3.10b) minimizes the vertical positional change of a label (Criterion 3) in two consecutive frames F and F^{-1}. Similarly, the third term (3.10c) minimizes the horizontal positional change. The last term (3.10d) reflects Criterion 2. By observing the influence of various weights on the resulting layouts with the defined criteria from Section 3.1 in mind, we recommend using the weights of the terms as $w_{row} = 0.5, w_{dist} = 0.8, w_{row_\Delta} = 1.0, w_{offset} = 0.1$ and $w_{center} = 0.3$.

To fulfill Requirement 1, we define the following constraint

$$a_{\mathrm{x}}\left(\ell^{(i)}\right) - \ell_{\mathrm{o}}^{(i)} + \ell_{\mathrm{w}}^{(i)} \leq a_{\mathrm{x}}\left(\ell^{(j)}\right) - \ell_{\mathrm{o}}^{(j)} + M \cdot \left(1 - I_{\ell^{(i)}}^r\right) + M \cdot \left(1 - I_{\ell^{(j)}}^r\right), \tag{3.11}$$

where we define the order so that $a_{\mathrm{x}}(\ell^{(i)}) \leq a_{\mathrm{x}}(\ell^{(j)}) \wedge \ell^{(i)} \neq \ell^{(j)}$ and $\ell^{(i)}, \ell^{(j)} \in \mathcal{L}^F$. To restrict the vertical position of any given label ℓ in frame F, which is placed in row $p = \ell_r^{F^{-1}}$ in the preceding frame F^{-1}, we introduce the constraint defined as

$$I_\ell^{p-1} + I_\ell^p + I_\ell^{p+1} = 1. \tag{3.12}$$

Furthermore, requirement Requirement 2 is defined in a similar way as in Section 3.2.1. Finally, a label ℓ is allowed to occupy only a single row r. Therefore, we define this restriction as

$$\sum_{r \in R} I_\ell^r = 1. \tag{3.13}$$

3.4 Extensions

Both proposed methods can be easily extended by additional terms and constraints to customize the resulting layout. In this section, we describe an extension for visibility optimization based on prominence and alpha-blending extension for smooth transitions of labels.

3.4.1 Feature Prominence and Visibility Optimization

Typically, some features of the visualized scene are more important than the other ones. Therefore, we define a *prominence* φ to express the importance of a label ℓ corresponding with an anchor $a(\ell)$. The prominence can be defined by a compound of several (potentially weighted) attributes. For example, the prominence of a mountain peak may be defined as a compound of its elevation, isolation, topographical prominence, and distance from the current viewpoint. To illustrate the *compound prominence* in our application, we define it as the weighted-sum of peaks' elevation ε, distance from the current viewpoint d, and Google score ω derived from a number of search results

$$\ell_\varphi = w_\varepsilon \ell_\varepsilon + w_d(1 - \ell_\mathrm{d}) + w_\omega \ell_\omega. \tag{3.14}$$

All attributes are normalized into the range $[0, 1]$. The weights were experimentally chosen as $w_\varepsilon = 1.0$, $w_d = 0.2$, and $w_\omega = 0.8$.

In crowded visualizations (*e.g.*, mountain terrain or city skyline) with many features to be labeled, it is sometimes useful not to show all the possible labels to prevent cluttered and chaotic label layout. More prominent features are more likely to be labeled and visualized; on the other hand, less prominent features do not need to be labeled at all. Therefore, the definition of Constraint 3.13 from Section 3.3, and ditto Constraint 3.5 from Section 3.2, can be redefined as

$$\sum_{r \in R} I_\ell^r \leq 1, \tag{3.15}$$

such that the label ℓ or the interval label λ, respectively, does not have to be placed in any row r. Furthermore, the cost function c_3 from Section 3.3.1, and ditto c_1 from Section 3.2.1, can be extended by term e_1 defined as

$$e_1(F) = -\rho_{\textcircled{v}} \sum_{\ell \in \mathcal{L}^F} \sum_{r \in R} I_\ell^r \left(\ell_\varphi + V(\ell, n)\right), \tag{3.16}$$

where the $\rho_{\textcircled{v}}$ is a reward for keeping the label visible, ℓ_φ is a compound prominence, and V is a *visibility function* that prevents labels from rapid disappearing. The visibility function V is piecewise-defined as

$$V(\ell, n) = \begin{cases} 1 & f - f_\ell^{\textcircled{n} \to \textcircled{v}} < n \\ 0 & else, \end{cases} \tag{3.17}$$

where f is the index of the current frame, $f_\ell^{\textcircled{n} \to \textcircled{v}}$ is the index of frame in which the last change from an *invisible state* $\textcircled{n}$ ($\forall r \in R : I_\ell^r = 0$) to a *visible state* $\textcircled{v}$ ($\exists r \in R : I_\ell^r = 1$) has occurred, and n is the number of frames for that the label should stay visible. To illustrate e_1 in our application, we set $\rho_{\textcircled{v}} = 0.9$, $n = 20$. Similarly, to prevent labels from rapid changing from $\textcircled{n}$ to $\textcircled{v}$ we define the term e_2 as

$$e_2(F) = -\rho_{\textcircled{n}} \sum_{\ell \in \mathcal{L}^F} \left(1 - \sum_{r \in R} I_\ell^r\right) \left((1 - \ell_\varphi) + N(\ell, n)\right), \tag{3.18}$$

where the $\rho_{\textcircled{n}}$ is a reward for keeping the label invisible, ℓ_φ is a compound prominence, and N is an *invisibility function* piecewise-defined as

$$N(\ell, n) = \begin{cases} 1 & f - f_\ell^{\textcircled{v}\to\textcircled{n}} < n \\ 0 & else, \end{cases} \tag{3.19}$$

where $f_\ell^{\textcircled{v}\to\textcircled{n}}$ is the index of frame in that the last change from $\textcircled{v}$ to a $\textcircled{n}$ has occurred. The rest mimics the definition of e_1. To illustrate e_2 in our application, we set $\rho_{\textcircled{n}} = 0.1$, $n = 10$.

3.4.2 Smooth Label Transition

To prevent labels from popping in and out abruptly, we implement smooth alpha-blending. An extended labeling method is then denoted by the suffix Alpha (*e.g.*, OnlineTemporalAlpha). Let $\kappa_\ell^F \in [0, 1]$ be the *alpha value* of label ℓ in frame F. When a label ℓ is added to the scene, the alpha value is set to $\kappa_\ell^F = \kappa_\Delta^{\textcircled{v}}$, where $\kappa_\Delta^{\textcircled{v}}$ is a constant that defines the fade in speed. In the following frames $f, \ldots, f^{\textcircled{v}\to\textcircled{n}} - 1$, the κ_ℓ^F is increased by a *fade in* function such as

$$\kappa_\ell^F = \min\left(\kappa_\ell^{F^{-1}} + \kappa_\Delta^{\textcircled{v}}, 1\right), \tag{3.20}$$

where $\kappa_\ell^{F^{-1}}$ alpha value of label ℓ in the previous frame F^{-1}. In our application we use $\kappa_\Delta^{\textcircled{v}} = 0.1$ (*i.e.*, the label is fully visible in 10 frames). Similarly, when a label ℓ is removed from the scene, the κ_ℓ^F is decremented by a *fade out* function such as

$$\kappa_\ell^F = \max\left(0, \kappa_\ell^{F^{-1}} - \kappa_\Delta^{\textcircled{n}}\right), \tag{3.21}$$

where the $\kappa_\Delta^{\textcircled{n}}$ defines the fade out speed. In our application we use $\kappa_\Delta^{\textcircled{v}} = 0.2$ (*i.e.*, the label fully disappears in five frames). For simplicity, we use the linear fade in/out function.

Furthermore, to create a smooth transition of a label ℓ during the fade-out blending in the proposed OnlineTmeporal method, the position of corresponding anchor $a_{\mathrm{x}}(\ell)$ needs to be predicted because the disappearance of the anchor $a_{\mathrm{x}}(\ell)$ cannot be precomputed in advance. Therefore, we apply *linear extrapolation* to calculate the $a_{\mathrm{x}}^F(\ell)$ from the two consecutive frames F^{-2} and F^{-1} such as

$$a_{\mathrm{x}}^F(\ell) = a_{\mathrm{x}}^{F^{-2}}(\ell) + 2\left(a_{\mathrm{x}}^{F^{-1}}(\ell) - a_{\mathrm{x}}^{F^{-2}}(\ell)\right). \tag{3.22}$$

3.5 Results

We used GUROBI 9.0 with a C++ interface as an optimization solver for the OfflineTemporal as well as OnlineTemporal method. The solver applies several primal heuristics and a branch-and-cut algorithm with different types of cutting planes (*e.g.*, Gomory, MIR, StrongCG) to solve the MILP and MIQP problem [67]. In case of the minimization of MIP problems, the branch-and-cut algorithm (for more details see, *e.g.*, Bixby *et al.* [27]) keeps track of the *upper bound* and *lower bound*. The upper bound UB (also called *incumbent*) is an objective value of the best feasible solution found so far. On the other hand, the lower bound LB is a minimum objective value of the LP-relaxed solutions (*i.e.*, integral constraints on variables are relaxed) in the leaf nodes of the branching tree. The absolute

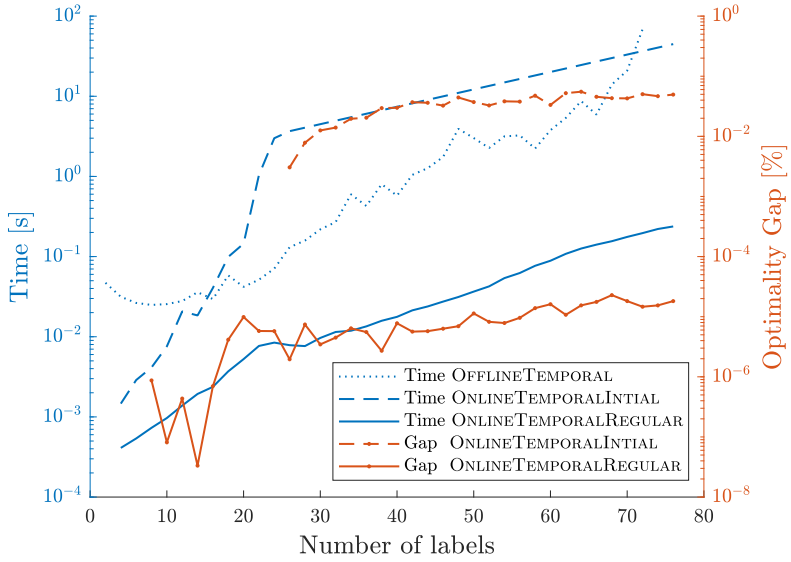


Figure 3.4: Time (left y-axis) and relative optimality gap (right y-axis) measurements for the proposed methods considering the number of labels in the scene.

difference between upper and lower bound serves as a quality measure of the solution to optimality. The optimization is terminated when the relative *optimality gap* G defined as

$$G = \frac{UB - LB}{UB} \cdot 100 \tag{3.23}$$

is less than the G_T % (the value of $G_T = 0.01$ % is, in fact, the recommended termination criterion by the authors of GUROBI), meaning that the solver found a near-optimal solution that can not be further than the G_T % from the true-optima. Note that, since the MILP, as well as MIQP, is $\mathcal{NP}$-hard, only exponential-time algorithms are known, and the computational time can grow significantly with an increasing number of binary variables (*i.e.*, with the number of labels) [28, 56]. The following measurements were performed on Intel® Core i7-9700K @ 3.60GHz with 64GB of RAM.

3.5.1 Offline Labeling Method

The solution of the *label interval to row assignment* for the sequence with 40 labels takes 575ms and for smaller instances (20 labels and less) it is found in less than 42ms (see Figure 3.4) with a relative optimality gap less than $G_T = 0.01$ %. The reported time measurements and optimality gaps are averaged over 100 runs.

The optimization in the *within row label placement* is defined as convex QP; hence it can be solved in polynomial time [184]. Moreover, the label placement can be solved independently for each row; therefore, the optimization is prompt and can run in parallel.

3.5.2 Online Labeling Method

The computation of the proposed OnlineTemporal method can be split into two phases. The solution for the *initial frame* F_0 is largely dependent on the search space, which is given by the number of labels. Each label ℓ can be placed at any row $r \in R$. The time needed to

solve the initial frame is a period of time when the interaction is not possible; thus, one has to wait until the solution is found. On the other hand, the search space for the following *regular frames* $F_1, F_2, \ldots, F_n$ is narrowed by the Constraint 3.12 (*i.e.*, the vertical change of any given label must be at most a single row). Therefore, label ℓ can be placed at any row ${\ell_r}^{F^{-1}} + c$ where the $c \in \{-1, 0, 1\}$. To limit the duration of the optimization for the initial frame, we apply the time restriction

$$t_{limit} = \exp\left(0.05 \cdot |\mathcal{L}|\right), \tag{3.24}$$

which in turn can increase the relative optimality gap G. The solution for the initial frame F_0 with 40 labels takes 7.4s with a relative optimality gap of $G = 0.03$ %. The solution of smaller instances (20 labels or less) is found in less than 145ms with a relative optimality gap of $G \approx 0$ %. To limit the duration of the optimization for the regular frames, we apply the time restriction

$$t_{limit} = \log\left(1 + 0.005 \cdot |\mathcal{L}|\right). \tag{3.25}$$

The solution for regular frames with 40 labels takes 18ms with a relative optimality gap of $G = 7.7 \times 10^{-6}$ %. The solution of smaller instances (20 labels or less) is found in less than 5ms with a relative optimality gap ranging from 3.33×10^{-8} to 9.87×10^{-6} %. The reported time measurements and optimality gaps are averaged over 100 runs. For more details see Figure 3.4.

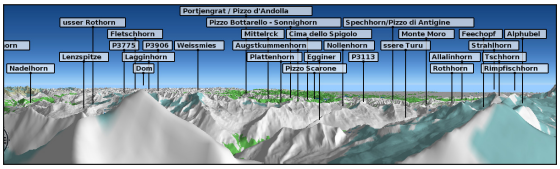
3.6 Comparison with State of the Art

We have implemented three previously published methods GrowingBorder [117], IntervalSlot [117], and GemsaMinRow [60] to compare them with the proposed OfflineTemporal and OnlineTemporal methods. The label layouts produced with these methods are shown in Figure 3.5(a)–3.5(f).

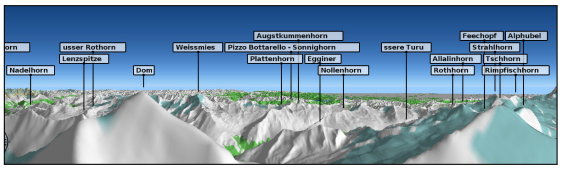
The GrowingBorder and IntervalSlot methods [117] were designed for the annotation of dynamic virtual landscapes. They connect each label to its vertical leader at a port in the center of the bottom boundary of the label. Due to this consistency, each label changes only its relative vertical position to its anchor. The relative horizontal positions of each label to its anchor is always the same (*i.e.*, only the length of the leader changes). This makes the movement of labels temporally coherent. The consistency can also facilitate finding the corresponding label to the given anchor (and vice versa). However, due to this fact, the methods may produce label layouts with longer leaders. Furthermore, they allow the leaders to intersect with labels of other anchors. Note that the latter two features can make finding the corresponding label to the given anchor (and vice versa) harder.

The GemsaMinRow method [60] was not intended for the annotation of dynamic scenes. Therefore, we apply the method to each frame independently. We do not expect the method to achieve a temporally coherent movement of labels. The method produces label layouts where the leaders do not intersect with any label, but create clusters of long leaders due to this constraint; see Figure 3.5(f). We have included the method into the comparison to examine whether the intersections of leaders with labels influence the users' ability to find the corresponding label to the given anchor (or vice versa).

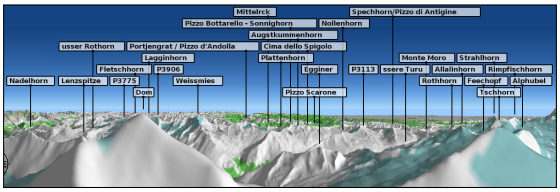
All compared methods were evaluated on three different sequences S_1, S_2, S_3 with a minimum length of 101 frames. The sequences are created by a series of horizontal and vertical movements that simulate, *e.g.*, the flight of a drone. Sequence S_1 consists of a long left *truck* (a leftwards horizontal movement of a camera) with a close anchor in front



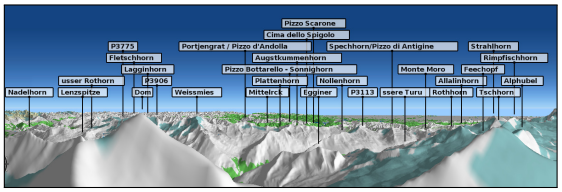
(a) Proposed OnlineTemporal method



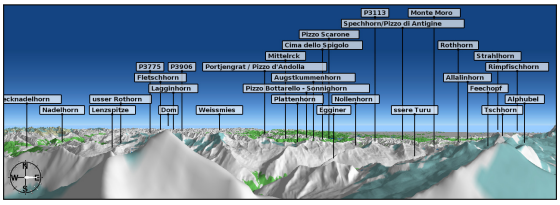
(b) Proposed OnlineTemporal method with extensions



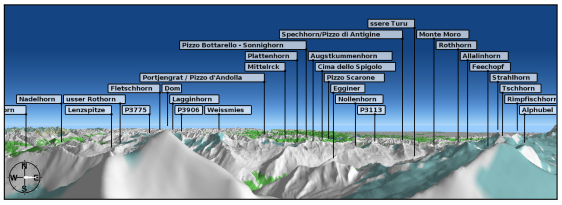
(c) Proposed OfflineTemporal method



(d) IntervalSlot method



(e) GrowingBorder method



(f) GemsaMinRow method

Figure 3.5: Example of label layouts calculated for the mountain peaks in the test sequence S_3 approximated by point anchors.

followed by a short *pedestal* down (a downwards vertical movement of a camera) where the anchors start to disappear rapidly. Sequence S_2 is composed as a sequence of a left truck, followed by a *dolly in* (a forward movement of a camera) completed by a pedestal up where the anchors start to appear rapidly. Sequence S_3 is created by a long right truck. Pan and tilt movements are not included as they do not introduce a parallax effect; therefore, the labeling does not change a lot, and it remains almost the same except for the newly appeared anchors on the edges of the drawing region D. Nevertheless, pan and tilt movements are included in the interactive experiment in Section 3.6.3.

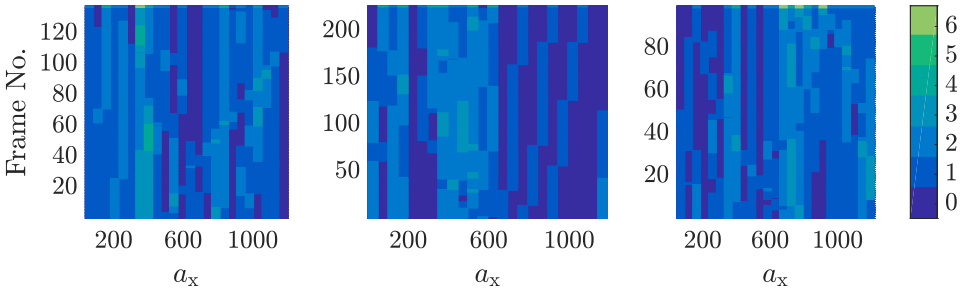
To further describe the sequences, we measured several anchor related parameters and characteristics for each sequence, please see Table 3.1 and Figure 3.6. The mean number of anchors within sequence S_1 and S_2 is 27 anchors, whereas, in the sequence S_3, it is only 21 anchors. The most noticeable changes concerning the anchors' x-position happen in the sequence S_1 (with the maximum shift of 31.97 pixels) and S_3 (with the maximum shift of 8.98 pixels). Regarding the anchors' y-position, only the anchors in the sequence S_1 and S_2 move dramatically. The length of an anchor interval reflects the distance that the anchor travels from the point it appears to the point it disappears. Allow us to point out that this also defines the space that is allocated for the smooth and uninterrupted movement of its label. Therefore, as the *Anchor Interval Length* section within Table 3.1 shows, in the sequence S_1 and S_3 exists an anchor that (a) moves very quickly in comparison to the other anchors, and (b) whose label interval allocates approximately a half of a row in a label layout. On average, an anchor is present in 98 out of 139 frames (71 %) in the sequence S_1, and more anchors disappear than appear throughout the sequence. The other two sequences S_2 and S_3 follows almost the same presence of 69 % and 67 %, respectively.

Parameters	Sequence		
	S_1	S_2	S_3
Number of Frames	139	228	101
Frame Dimensions	1200×400	1200×400	1281×346
Number of Anchors	*anchors*		
Minimum	21	19	26
Mean	27	21	27
Maximum	30	24	30
Anchor Shift in Frame	*pixels (x, y)*		
Minimum	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)
Mean	(0.36, 0.32)	(0.20, 0.19)	(0.54, 0.19)
Maximum	(31.97, 5.51)	(1.66, 2.40)	(8.98, 0.86)
Anchor Interval Length	*pixels*		
Minimum	1.20	0.10	0.90
Mean	38.40	30.68	44.48
Maximum	663.73	159.15	534.69
Anchor Presence	*no. of frames*		
Shortest	7	6	9
Mean	98	158	68
Longest	139	228	101
Anchor Presence Change	*no. of anchors*		
Appeared	19	18	23
Disappeared	24	19	22
Anchor Clusters	*no. of clusters (bin size 50 px)*		
Size 6	1	0	3
Size 4	32	6	3
Size 3	171	144	132
Size 2	754	1445	613

Table 3.1: Parameters of test sequences. Mean *Anchor Shift in Frame* is calculated from individual mean shifts of anchors over a sequence. *Anchor Interval Length* is equivalent to an absolute change of anchor's x-position.

The density maps of anchors' x-coordinates (a_x) depicted in Figure 3.6 reveals that (with respect to the bin size of 50 px, $D_w \approx 1200$ px) in sequence S_1, there is one cluster of six anchors, 32 clusters of four anchors, 171 clusters of three anchors and 754 clusters of two anchors. For more details about sequences, please see Table 3.1 and Figure 3.6. In addition, the tested sequences are available online as a part of Appendix A.

For all compared methods, we evaluate if the label layouts produced for the sequences S_1, S_2, and S_3 are temporally coherent and if the label layouts allow users to find the corresponding label to the given anchor (or vice versa). Furthermore, we evaluate users' preferences among all compared methods.



(a) Sequence S_1 (b) Sequence S_2 (c) Sequence S_3

Figure 3.6: Density of anchors' x-coordinates (a_x) over frames of test sequences S_1 (a), S_2 (b) and S_3 (c). The anchors are accumulated into bins with a width of 50 pixels.

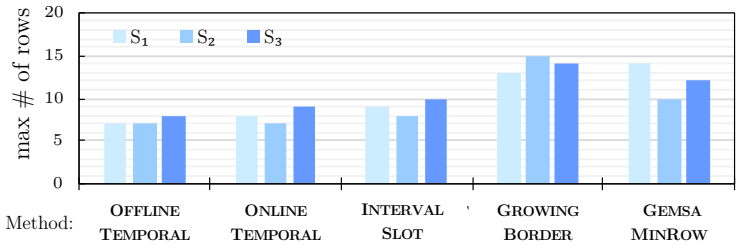
3.6.1 Quantitative Evaluation

In the quantitative evaluation, we have measured properties (vertical compactness, vertical displacement, and horizontal displacement) of the produced label layouts for the sequences S_1, S_2, and S_3 for each of the compared methods. The vertical compactness of the label layout can be described by the *maximum number of rows* M_r in a sequence. The methods producing label layouts with lower M_r are able to position the same number of labels on a lower number of rows. The compactness is important as the labels must fit into a labeling region L of finite height. Furthermore, label layouts with high M_r will lead to long leaders that can make finding the corresponding label to the given anchor (and vice versa) harder. The results presented in Figure 3.7(a) show that the OfflineTemporal method, followed by the OnlineTemporal method, achieves the best results. Note that the GrowingBorder and GemsaMinRow methods achieve the worst results leading to longer leaders.

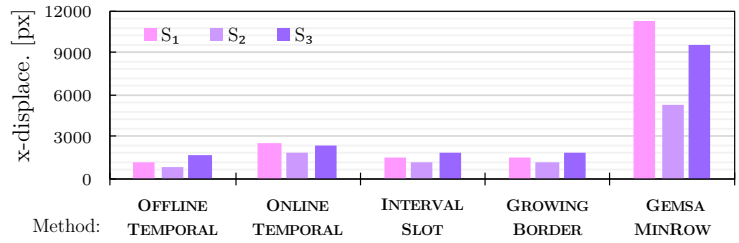
For the temporally coherent movement of labels, the labels must not jump abruptly. Therefore, we have calculated the *displacement metric* for the compared methods in horizontal Δ_x and vertical Δ_y direction separately as the sum of differences in the positions of all labels between all pairs of subsequent frames. The results presented in Figure 3.7(b) and Figure 3.7(c) suggest that labels in our proposed methods are more temporally coherent than in the other compared methods. The most significant discrepancy is visible in horizontal displacement Δ_y. The GemsaMinRow method achieves the worst results for both horizontal displacement Δ_x and vertical displacement Δ_y. This is expected as the method was not designed for the labeling of dynamic scenes.

3.6.2 Accuracy Experiments

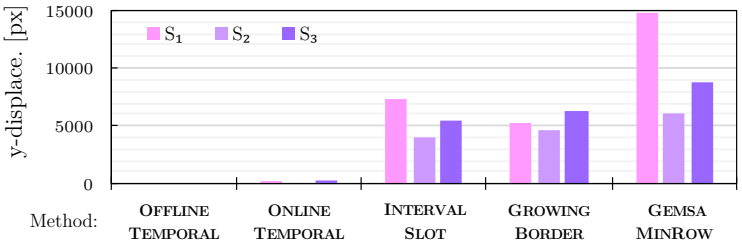
We have conducted a user study to assess whether the proposed OfflineTemporal and OnlineTemporal methods (1) improve the ability of the user to follow the labels in time and (2) influence the ability of the label layout to mediate the interconnection between the labels and the features. For the evaluation, we have created a web application that the participants accessed through a web browser. First, each participant was instructed about the testing procedure; then, the participant provided their age and gender. The evaluation was divided into two experiments. The first experiment was one factor with four levels. The independent variable was the labeling method. The four levels were our OfflineTemporal method and IntervalSlot, GrowingBorder, and GemsaMinRow



(a) Maximum number of rows M_r



(b) Displacement Δ_x in horizontal direction



(c) Displacement Δ_y in vertical direction

Figure 3.7: Quantitative metrics: (a) The maximum number of rows per sequence of the label layout. Total label displacement per sequence in horizontal (b) and vertical (c) direction. The Δ_y for the OfflineTemporal method is zero. The tested sequences S_1, S_2, S_3 are represented by individual colors.

methods. The follow-up experiment was one factor with one level. Again, the independent variable was the labeling method. The only level was the OnlineTemporal method. In both experiments, we evaluated the methods for three sequences S_1, S_2, S_3.

Both experiments were designed as a between-subject. In other words, one participant was tested with only one labeling method to eliminate the learning effect and fatigue. For each participant, the order of sequences was counterbalanced with a 3x3 balanced Latin square [118, Section 5.11] to eliminate the carry-over effect. In a between-subject design, combining the results of the two experiments is trivial as each level is evaluated independently of the other levels. Both experiments consisted of a series of three tasks defined as follows:

Task 1. Locate the label associated to a highlighted anchor.

Task 2. Locate the anchor associated to a highlighted label.

Task 3. Follow a moving label for two seconds and then select the label in the blind view (*i.e.*, the text of the label is not shown).

For a detailed description of the tasks, please see Appendix A and supplementary video[1].

Each participant repeated each task 10 times for each sequence. We measured the *error rate* (the number of wrongly selected labels/anchors relative to all selected labels/anchors). Afterward, we conducted a subjective evaluation of the visual search easiness (Task 1–3), the confidence (Task 1–2) and the need to focus (Task 3). The participants provided their subjective evaluation on Likert scales from 1 to 5.

Task 1 and its subjective evaluation was completed by 60 participants (12 females) with the age ranging from 19 to 54 years ($\bar{x} = 25.31$; $\sigma = 6.49$) in the first experiment and by 35 participants (three females) with the age ranging from 20 to 38 years ($\bar{x} = 24.03$; $\sigma = 4.62$) in the second experiment. Task 2 and its subjective evaluation was completed by 49 participants (11 females) with the age ranging from 19 to 54 years ($\bar{x} = 25.86$; $\sigma = 7.04$) in the first experiment and by 25 participants (two females) with the age ranging from 20 to 38 years ($\bar{x} = 23.92$; $\sigma = 4.76$) in the second experiment. Finally, Task 3 and its subjective evaluation was completed by 44 participants (10 females) with the age ranging from 19 to 54 years ($\bar{x} = 26.32$; $\sigma = 7.29$) in the first experiment and by 24 participants (two females) with the age ranging from 20 to 38 years ($\bar{x} = 23.92$; $\sigma = 4.86$) in the second experiment.

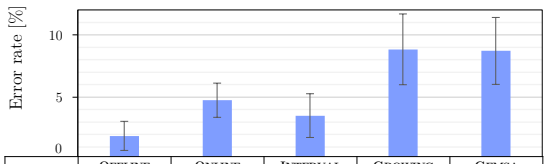
For each task and each measured variable, we define a family of two null hypotheses – H_0^1: "The difference in the mean value of the OfflineTemporal method and the mean value of each other compared method is zero," and H_0^2: "The difference in the mean value of the OnlineTemporal method and the mean value of each other compared method is zero." The family of hypotheses consists of 7 pairwise comparisons.

We evaluated the collected data for all sequences together. We performed a statistical evaluation of the measured data using confidence intervals. We calculated the confidence intervals of the error rates as adjusted Wald intervals, a method recommended for completion rates [2, 149]. We calculated the confidence intervals for Likert scales as confidence intervals for rating scales [150, Chapter 3]. We used 95% confidence intervals for error rates and Likert scales.

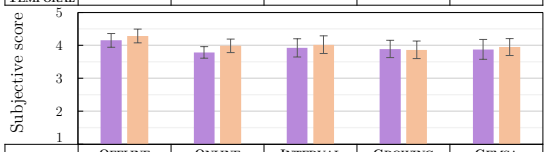
For Task 1 and Task 2, the average error rates and average score from subjective evaluation together with their 95% confidence intervals are shown in Figure 3.8(a) and Figure 3.8(b). For Task 3, the average error rates, and average scores from the subjective evaluation, together with their 95% confidence intervals, are shown in Figure 3.8(c). Please note that the 95% confidence intervals cannot be directly used to visually evaluate the difference between the means if multiple pairwise caparisons are evaluated.

To detect whether the means of the measured data are significantly different, we have calculated [5] the p-value from the 95% confidence interval of the difference between the means for all seven pairwise comparisons. To keep the type 1 error at significance level $\alpha = 0.05$ for the whole family of hypotheses, we have adjusted the p-values with the Holm's [78] sequentially-rejective method using the Šidák equation [155] as described by Ludbrook [116]. Please note that in certain cases, the method produces the same adjusted p-values for several pairwise comparisons. We report the adjusted p-values in Figure 3.8. As we are adjusting the p-values, not the significance level α, we compare all adjusted p-values with $\alpha = 0.05$.

[1]Supplementary video is available at the project page http://cphoto.fit.vutbr.cz/interactive-labeling/

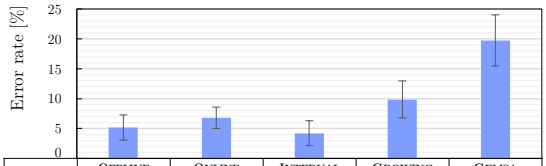


Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Error rate	1.89(0.73, 3.06)	4.75(3.38, 6.13)	3.52(1.77, 5.27)	8.83(6.0, 11.64)	8.71(6.03, 11.4)
Offline Temporal	—	$p = 0.0095$	$p = 0.2439$	$p < 0.0001$	$p < 0.0001$
Online Temporal	—	—	$p = 0.2810$	$p = 0.0397$	$p = 0.0397$

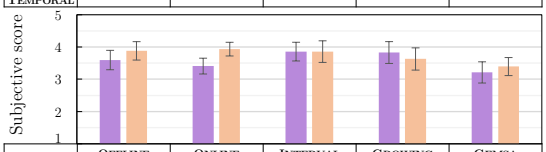


Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Easiness	4.15(3.95, 4.36)	3.79(3.61, 3.97)	3.93(3.65, 4.20)	3.90(3.63, 4.16)	3.88(3.58, 4.18)
Offline Temporal	—	$p = 0.0602$	$p = 0.5929$	$p = 0.5690$	$p = 0.5690$
Online Temporal	—	—	$p = 0.8023$	$p = 0.8023$	$p = 0.8023$
Confid.	4.29(4.08, 4.50)	3.99(3.78, 4.19)	4.02(3.76, 4.29)	3.87(3.60, 4.14)	3.95(3.70, 4.21)
Offline Temporal	—	$p = 0.2386$	$p = 0.4199$	$p = 0.2386$	$p = 0.2809$
Online Temporal	—	—	$p = 0.9733$	$p = 0.8672$	$p = 0.9733$

(a) Task 1: Assign label to highlighted anchor

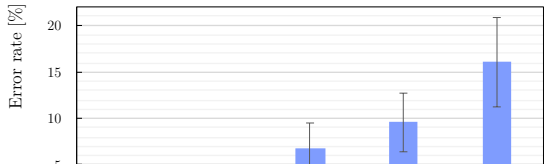


Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Error rate	5.17(3.06, 7.28)	6.8(5.01, 8.58)	4.22(2.12, 6.31)	9.87(6.76,12.98)	19.75(15.48, 24)
Offline Temporal	—	$p = 0.4399$	$p = 0.5397$	$p = 0.0688$	$p < 0.0001$
Online Temporal	—	—	$p = 0.2380$	$p = 0.2531$	$p < 0.0001$

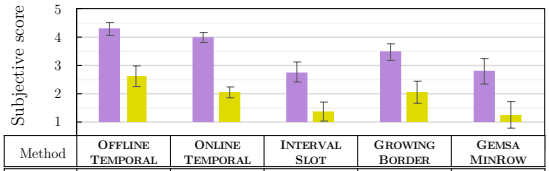


Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Easiness	3.60(3.30, 3.90)	3.41(3.16, 3.65)	3.86(3.57, 4.15)	3.83(3.49, 4.17)	3.21(2.88, 3.54)
Offline Temporal	—	$p = 0.6793$	$p = 0.6305$	$p = 0.6793$	$p = 0.3811$
Online Temporal	—	—	$p = 0.1337$	$p = 0.2562$	$p = 0.6793$
Confid.	3.88(3.60, 4.17)	3.93(3.72, 4.15)	3.86(3.52, 4.19)	3.63(3.28, 3.97)	3.39(3.11, 3.67)
Offline Temporal	—	$p = 0.9772$	$p = 0.9772$	$p = 0.7218$	$p = 0.0976$
Online Temporal	—	—	$p = 0.9772$	$p = 0.5299$	$p = 0.0183$

(b) Task 2: Assign anchor to highlighted label



Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Error rate	0.71(-0.10, 1.51)	2.41(1.31, 3.51)	6.77(4.03, 9.51)	9.56(6.41, 12.72)	16.05(11.24,20.86)
Offline Temporal	—	$p = 0.0145$	$p = 0.0001$	$p < 0.0001$	$p < 0.0001$
Online Temporal	—	—	$p = 0.0076$	$p = 0.0001$	$p < 0.0001$



Method	Offline Temporal	Online Temporal	Interval Slot	Growing Border	Gemsa MinRow
Easiness	4.28 (4.05, 4.50)	3.97 (3.8, 4.15)	2.75 (2.40, 3.11)	3.46 (3.16, 3.75)	2.77 (2.32, 3.23)
Offline Temporal	—	$p = 0.0401$	$p < 0.0001$	$p < 0.0001$	$p < 0.0001$
Online Temporal	—	—	$p < 0.0001$	$p = 0.0065$	$p < 0.0001$
Focus	2.60 (2.23, 2.97)	2.03 (1.84, 2.22)	1.34 (1.01, 1.68)	2.03 (1.64, 2.42)	1.23 (0.76, 1.70)
Offline Temporal	—	$p = 0.0204$	$p < 0.0001$	$p = 0.0729$	$p < 0.0001$
Online Temporal	—	—	$p = 0.0030$	$p = 0.9903$	$p = 0.0085$

(c) Task 3: Follow the moving label

Figure 3.8: Results of the Accuracy Experiment: Error rate and subjective score for the Task 1 (a), Task 2 (b), and Task 3 (c).

Task 1: Assign Label to Highlighted Anchor

The results in Figure 3.8(a) show that the OfflineTemporal method achieves a significantly lower error rate than the OnlineTemporal, GrowingBorder, and GemsaMinRow methods. The OnlineTemporal method achieves a significantly lower error rate than the GrowingBorder and GemsaMinRow methods. In the subjective evaluation, we have not detected any significant difference for the easiness and for the confidence. The results indicate that the label layouts with longer leaders (GrowingBorder and GemsaMinRow) negatively influence the ability of users to assign the correct label to the selected anchor. The leaders that do not intersect the labels (GemsaMinRow) do not compensate for the longer leaders.

Task 2: Assign Anchor to Highlighted Label

The results in Figure 3.8(b) show that the OfflineTemporal and OnlineTemporal methods achieve a significantly lower error rate than the GemsaMinRow method. In the subjective evaluation of Task 2, we have not detected any significant difference for easiness. However, the participants were significantly more confident that they are selecting the correct anchor with the OnlineTemporal method than with the GemsaMinRow method. In our opinion, the clusters of long leaders created by the GemsaMinRow method are the reason for the poor performance of the method.

Task 3: Follow the Moving Label

The results in Figure 3.8(c) show that the OfflineTemporal method achieves a significantly lower error rate than all other methods. The OnlineTemporal method achieves a significantly lower error rate than the IntervalSlot, GrowingBorder and GemsaMinRow methods. In the subjective evaluation of Task 3, the participants reported that the task was significantly easier to complete with the OfflineTemporal method than with all other methods. The task was significantly easier to complete with the OnlineTemporal method than with the IntervalSlot, GrowingBorder, and GemsaMinRow methods. Furthermore, the participants reported that they had to focus significantly less with the OfflineTemporal method than with the the OnlineTemporal, IntervalSlot, and GemsaMinRow methods. With the OnlineTemporal method they had to focus significantly less than with the IntervalSlot and GemsaMinRow methods. The results strongly indicate that lower displacement of labels between the frames improves the accuracy of the users in following a moving label.

Discussion

In general, the results show that for the three sequences S_1, S_2, S_3 our proposed OfflineTemporal method followed by the OnlineTemporal method allow to follow labels moving in time significantly more accurately than the compared methods. At the same time, our OfflineTemporal and OnlineTemporal methods mediate the interconnection between labels and anchors the same as (IntervalSlot for Task 1 and Task 2 and GrowingBorder for Task 2) or better than (GemsaMinRow for Task 1 and Task 2 and GrowingBorder for Task 1) the compared methods. We were especially surprised by the poor performance of the GemsaMinRow method in Task 1 and Task 2. It seems that forcing the leaders not to intersect with labels is counterproductive as it leads to clusters of long leaders. In conclusion, we recommend using the proposed methods over the compared methods for sequences with similar characteristics as the sequences S_1, S_2, S_3.

3.6.3 Preference Experiments

To assess the users' preferences among different labeling methods, we have conducted two subjective experiments. The first was designed as non-interactive (*i.e.*, participants could not influence the pose of the camera in the scene), and the second as interactive (*i.e.*, participants were asked to interact with the camera in the scene).

Non-Interactive Environment

To capture users' preferences in a non-interactive environment, we conducted an experiment based on a psychophysical technique of paired comparisons [45, 166]. Specifically, we exploited the two-interval forced choice (2IFC) paradigm to verify the perceived quality of labeling methods (GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal, OnlineTemporal, and its extended version by alpha-blending denoted as OnlineTemporalAlpha).

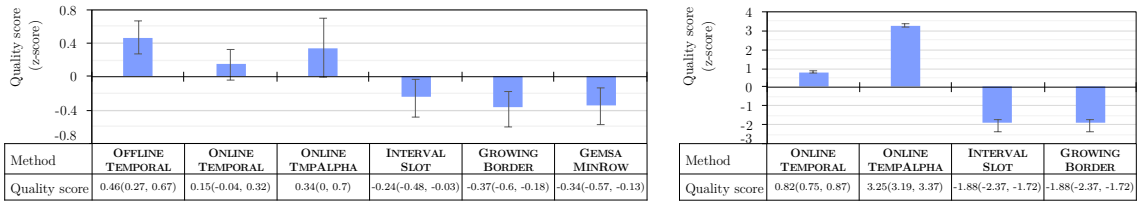
At the beginning of the experiment, participants were familiarized with the experimental procedure by the written instructions. Participants were asked to focus on the visual presentation of the labels and then select the method they liked the most. During the experiment, participants were able to play the assigned sequence as many times as they wanted. The names of the methods were transcoded with numbers. The stimuli were represented by three different video sequences presented in a web browser, and we evenly distributed them among the participants. Each participant was sequentially stimulated by a pair of labeling methods applied to the assigned sequence.

We have collected two groups of participants A and B, each consisting of 40 persons. The participants in group A — 40 males and 10 females with the age ranging from 19 to 54 years ($\bar{x} = 26.61$; $\sigma = 7.46$) — were asked to compare all pairs of GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal. Therefore, each participant in group A contributed with $\binom{m}{2} = 6$ pairwise comparisons where $m = 4$. Moreover, the order of the pairs of methods to compare was counterbalanced with a 6x6 balanced Latin square [118, Section 5.11] to eliminate learning and carry-over effects. Based on the outlier analysis tool provided by Pérez-Ortiz and Mantiuk [139], four participants (two males, two female) that behave very differently from the others were removed. The participants in group B — 36 males and four females with the age ranging from 17 to 44 years ($\bar{x} = 26.18$; $\sigma = 5.76$) — were asked to compare randomized pairs of {GemsaMinRow, GrowingBorder, IntervalSlot, OfflineTemporal} × OnlineTemporal extended by a pair of OnlineTemporal × OnlineTemporalAlpha which was presented as the last pair of the experimental procedure. Therefore, each participant in group B contributed with five pairwise comparisons. Together, group A and B created an incomplete experimental design (*i.e.*, only several pairs are compared) to reduce the number of needed pairwise comparisons.

We stored the data in the count matrix $\mathbf{C}$ for each participant separately. The element c_{ij} represents the number of times that method i was selected over method j. We converted the per-participant-count matrices $\mathbf{C}$ into a quality score (z-score) scale and computed a statistical significance using a customized MATLAB framework [139]. To transform the count matrix $\mathbf{C}$ to the quality score scale, we used Thurstone's Law of Comparative Judgment model concerning Case V [139, 166]. In order to reject the null hypothesis H_0^3: "the difference in perceived quality scores is zero," we applied the Two-tailed test at a significance level of $\alpha = 0.05$.

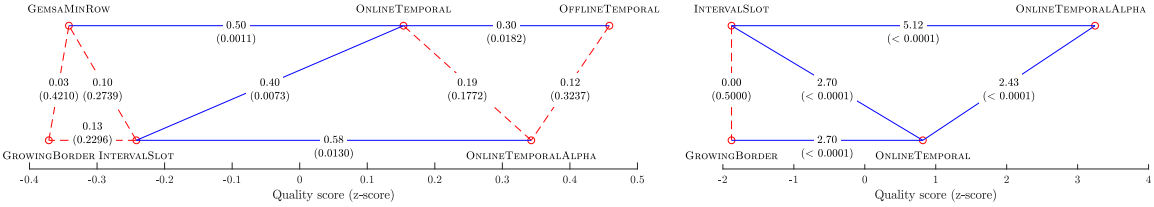
The quality scores for compared methods are depicted in Figure 3.9(a). The results show that the proposed OfflineTemporal method exhibit the best quality score followed by OnlineTemporalAlpha and OnlineTemporal. The results also suggest that the best of the previously published methods is considered to be IntervalSlot followed by GemsaMinRow and GrowingBorder.

The statistical significance for surveyed methods is presented in Figure 3.10(a). The quality difference between the proposed OfflineTemporal and OnlineTemporal is



(a) (b)

Figure 3.9: Quality scores and 95% confidence intervals for (a) non-interactive and (b) interactive experiment.



(a) (b)

Figure 3.10: Statistical significance (p-values reported in brackets) and quality scores for (a) non-interactive and (b) interactive experiment.

statistically significant. Therefore, we can reject the null hypothesis H_0^3 for the pairs of OFFLINETEMPORAL × {ONLINETEMPORAL, INTERVALSLOT, GROWINGBORDER, GEMSAMINROW}. The ONLINETEMPORALALPHA has a higher quality score ($q = 0.34$) than ONLINETEMPORAL ($q = 0.15$). However, H_0^3 can not be rejected for this pair and more generally for any other pair of {OFFLINETEMPORAL, ONLINETEMPORAL} × ONLINETEMPORALALPHA. Therefore, the suggested additional quality of alpha-blending is not statistically proved. In addition, we have not detected significant difference in perceived quality among the INTERVALSLOT, GROWINGBORDER and GEMSAMINROW methods.

Interactive Environment

To capture users' preferences in the nature of an interactive environment, we conducted a follow-up experiment. Participants had to rank the labeling methods suitable for interactive applications (GROWINGBORDER, INTERVALSLOT, ONLINETEMPORAL and ONLINETEMPORALALPHA) from the best perceived method to the worst. We transcoded the names of the methods with numbers, and the first shown method was randomized. At the beginning of the experiment, participants were familiarized with the experimental procedure by the written instructions. We instructed the participants to focus predominantly on the assessment of label placement and the movement of the labels in time. After they read the instructions, the supervisor again repeated all the important details. The stimuli were represented by an interactive visualization of mountain terrain presented at a resolution of 1200x900, where the independent variable was the labeling method. During the experiment, participants were guided along the same predefined path above mountain peaks. They could interact with the scene by the following operations whenever they wanted: (1) fly forward/backward and stop, (2) rotate the camera, (3) zoom in and out, (4) return to the beginning of the path, and (5) change the labeling method. For convenience, we provided participants with

printed cards to ease the ranking $\mathbf{R}$ during the experiment. At the end of the experiment, participants were asked to describe their decision process and to justify their ranking.

A total of 15 participants (two females) with the age ranging from 22 to 35 completed the experiment in $\bar{t} = 14$ minutes ($\sigma = 3.5$). Based on the described decision process and justification description, we removed three participants (two males, one female) that were wrongly focused on the other aspects of the presented stimulus (*e.g.*, the length of leaders, the rendering method, or correspondence of the label's row with the distance from the camera). The removed participants were also suggested by the outlier analysis tool provided by Pérez-Ortiz and Mantiuk [139].

To be consistent with the previous experiment, we chose once again to apply the pairwise comparison paradigm. We used the transitive closure to transform the ranking $\mathbf{R}$ to the count matrix $\mathbf{C}$. For example, the participant's ranking A, B, C was transformed to a pairwise comparison $[A, B], [B, C], [A, C]$. Afterward, we derived the quality score by Thurstone's Law of Comparative Judgment model concerning Case V [139, 166]. In order to reject the null hypothesis (the same as H_0^3 from Section 3.6.3): "the difference in perceived quality scores is zero," we applied the Two-tailed test at a significance level of $\alpha = 0.05$.

The quality scores for the compared methods is depicted in Figure 3.9(b). The results show that the proposed methods — OnlineTemporalAlpha, OnlineTemporal — exhibit the best quality score followed by IntervalSlot and GrowingBorder. The statistical significance for the surveyed methods is presented in Figure 3.10(b). The OnlineTemporalAlpha has a higher quality score ($q = 3.25$) than OnlineTemporal ($q = 0.82$). Unlike the previous non-interactive experiment, the results show that the quality difference between the proposed OnlineTemporal and OnlineTemporalAlpha is statistically significant. Therefore, we can reject the null hypothesis H_0^3, and the suggested additional quality of alpha-blending is, in this case, statistically proved. Furthermore, we can reject the null hypothesis H_0^3 for the pairs of {OnlineTemporal OnlineTemporalAlpha} $\times$ {IntervalSlot, GrowingBorder}. Consistently with the previous experiment, we have not detected a significant difference in perceived quality between the IntervalSlot and GrowingBorder.

3.7 Limitations

The proposed methods are seemingly more involved and harder to implement than the compared state-of-the-art methods. However, the labeling approached as an optimization problem yields a higher level of flexibility, which allows extending the proposed formulations for diverse needs. Furthermore, the difference between the lower and upper bounds used to solve MIP problems provide a quality measure of the solution to optimality. Another drawback of our approaches is the time-to-solve span for the initial frame which grows with an increasing number of labels in a scene. We tackled this issue by limiting the available time in favor of optimality; however, in future work, this could be approached differently (*e.g.*, by further narrowing the search space similar to the approach for regular frames).

3.8 Summary

In this chapter, we proposed two novel temporally stable screen-space methods for boundary labeling of dynamic scenes using an optimization approach. The OfflineTemporal

method is designed for the offline processing of the dynamic scene in advance (*e.g.*, drone-shot videos annotation, visualizations in television news). On the other hand, the OnlineTemporal method is designed for interactive applications (*e.g.*, terrain viewers, augmented, and virtual reality applications). Both proposed methods can be easily extended by additional terms and constraints to customize the resulting label layout, such as visibility optimization based on prominence and alpha-blending extension for smooth label transition. We show that according to the results of quantitative evaluation, the label layout is as compact as previous methods. At the same time, labels are more stable during an interaction with the scene. Furthermore, we compared the methods with three previously published methods in an extensive user study. The results of the accuracy experiment show that with our methods, the users can follow moving labels significantly more accurately than with the concurrent methods. At the same time, our methods mediate the interconnection ability between labels and features the same as or better than the other methods. Moreover, the results of the preference experiment show that the proposed methods were ranked the best for both interactive and non-interactive boundary labeling of dynamic scenes.

Chapter 4

Deep Reinforcement Learning Approach to Internal Labeling

Just as deep learning vastly mitigated the need for feature engineering and produced remarkable performance improvements in computer vision and natural language processing, applying learning techniques to visualization could lead to similar advancements.

In this chapter, we present point-feature label placement as a *reinforcement learning* problem. Reinforcement Learning (RL) is an area of machine learning (alongside supervised and unsupervised learning) concerned with decision-making, driven by experience, to maximize the numerical reward signal. Over the discrete-time steps, an agent (*i.e.*, decision-maker) senses the *state* of the *environment*, interacts with the environment by taking *actions* that affect the state, and receives a *reward*. Learning from experience overcomes the lack of label placement datasets that would be needed for supervised learning. Furthermore, the advancement in Deep Learning and its combination with RL emerged in Deep Reinforcement Learning (DRL). DRL has enabled scaling the RL to previously intractable problems (including labeling) due to the curse of dimensionality. Recent research has successfully proven that DRL can solve complex problems within several domains even with enormous state spaces, *e.g.*, robotics [103, 104], self-driving cars [136], industrial design [125], and finance [46]. Moreover, DRL can be trained for objectives that are difficult to optimize directly, as DRL is agnostic to the precise model of the environment as long as the reward signal correlates with the objective. Li and Malik [106] demonstrated that an RL-based autonomous optimization algorithm converges faster and/or finds better optima than the existing hand-engineered optimization algorithms (*e.g.*, gradient descent, momentum, conjugate gradient).

Given the mentioned characteristics, we believe the DRL approach is well suited for the point-feature label placement problem. However, employing RL to label placement poses unique challenges, such as complex deep RL training, handling variable number of anchors and labels, and vast continuous state and action spaces. Moreover, it demands the meticulous design of neural network architectures, formulation of state space, and careful definition of action space. Despite these challenges, the potential of RL in this domain is immense. Our research confronts these issues, devising strategies highlighting our novel approach to the problem.

To the best of our knowledge, no work has been published on label placement with the ability to learn and generalize from experience and generate new labeling for unseen sets of features of interest. Our main contributions are summarized as follows:

(**1**) We introduce a Multi-Agent Deep Reinforcement Learning formulation to the point-feature labeling problem, which we believe to be the first machine-learning-driven labeling method contrary to the existing hand-crafted algorithms designed by human experts. Our efficient feedforward neural network architecture, with less than half a million parameters, serves as both a policy and value function approximation.
(**2**) We provide comprehensive ablation studies on the observed modalities and neural network architecture that justify our design choices, highlight the essential aspects of our work, and provide a guideline for future work. As a part, we introduce a novel *completeness metric* that measures the performance of labeling methods based on the number of completely labeled point features and *benchmark dataset* for evaluation.
(**3**) We compare the performance of the proposed method with two existing methods, Particle-Based Labeling [115] and Rapid Labels [138], using quantitative assessments. Additionally, we conducted a user study to evaluate the proposed method qualitatively. Based on these evaluations, we show that our method outperforms all the other examined methods.

Machine Learning & Visualization

There has been a recent trend of incorporating machine learning techniques into visualization (ML4VIS) to enhance the efficiency of visualization creation and suggest appropriate visual representations. Machine learning can be broadly categorized into several types, including supervised learning and reinforcement learning, each with its unique characteristics and suitable application areas. Supervised learning is the most common type, where an algorithm learns a model from labeled training data and then uses this model to make predictions for unseen data. In the context of visualization, supervised learning has been utilized in several impactful ways. For instance, Luo *et al.* [114] introduced DeepEye, a system that leverages supervised learning to recommend suitable visualizations based on the provided data. Likewise, Bylinskii *et al.* [35] formulated a method for predicting users' visual attention distribution on infographics. Similarly, Chen *et al.* [38] employed a series of supervised-learning techniques for the automatic segmentation of graphical elements from timeline infographics.

Reinforcement learning represents another significant category of machine learning, where an agent learns to make decisions by taking actions in an environment to maximize some notion of cumulative reward. Despite the success of reinforcement learning across various domains, its application in visualization (RL4VIS) remains relatively unexplored, with only a handful of works adopting this technique. For instance, Tang *et al.* [162] proposed PlotThread, which facilitates collaboration between an RL agent and a human designer to modify the storyline's layout. The trained agent assists designers and refines their interactions on a shared canvas. Zhou *et al.* [189] proposed Table2Chart leveraging deep Q-learning and heuristic search to generate chart sequences from table data. Hu *et al.* [79] proposed a method for optimizing the coordinate ordering of sets of star glyphs related to multiple class labels to maximize perceptual class separation. In addition, Wu *et al.* [179] developed MobileVisFixer, which automates a mobile-friendly visualization re-design process, and Deng *et al.* [44] introduced DashBot, a Deep RL-based tool for generating analytical dashboards. For a more comprehensive review of the use of machine learning in visualization, please refer to the survey by Wang *et al.* [175]. While RL provides innovative solutions for visualization problems, visual analytics also proves instrumental in understanding and interpreting RL models (VIS4RL). For instance, DQNViz [173] proposed by Wang *et al.*

and DRLViz [84] introduced by Jaunet *et al.* offer visual analytics approaches to understand deep Q-networks and deep reinforcement learning, respectively. Mishra *et al.* [126] visualized explanations of agent behavior in reinforcement learning, whereas Wang *et al.* [174] employed visual analytics for RNN-based deep reinforcement learning.

Although the techniques above tackle various tasks, none of them address the specific challenge of label placement. Furthermore, RL-based techniques predominantly rely on a single-agent approach, a strategy that diverges from the multi-agent system proposed in our research. Our work introduces a novel approach wherein multiple agents, acting as proxies for labels, are tasked to learn and interact concurrently within a shared environment. This paradigm inherently involves more complex dynamics, such as agent coordination and management of a non-stationary environment, which arise due to the simultaneous learning processes undertaken by the agents [36, 113]. Therefore, our work represents not only an advancement in the application of reinforcement learning to the intricate problem of label placement but also the exploration of multi-agent systems in the visualization domain.

4.1 Introduction to Reinforcement Learning

Instead of relying on labeled data common for supervised learning, reinforcement learning (RL) leverages an agent that learns to take actions that maximize a cumulative reward by exploring the environment and observing the consequences of its actions. Reinforcement learning techniques have become an essential tool for solving various complex problems, particularly when dealing with decision-making. However, the curse of dimensionality in state space makes it challenging to apply RL effectively. To overcome this, RL is usually extended to Deep Reinforcement Learning (DRL), which employs a neural network as a function approximator [161]. Based on the number of agents, RL can be further classified into a Single-Agent Reinforcement Learning (SARL) variant, where only one agent interacts with the environment. The second variant is Multi-Agent Reinforcement Learning (MARL), where multiple agents interact with a shared environment and work collaboratively to attain a shared goal.

The formal basis of reinforcement learning is the theory of Markov Decision Processes (MDP). Over the discrete-time steps $t \in \mathbb{N}$, an agent observes the *state* $s_t \in S$ of the *environment*. Given the state, the agent selects *actions* $a_t \in A$ based on a *policy* π that maps the state to a probability distribution over actions. At the next time step $t+1$ as a consequence of the selected action, the agent receives feedback in a form of numerical *reward* r_{t+1} and transitions to new state s_{t+1}.[1] The sequence of states, actions, and rewards is called *trajectory* or *rollout*. The agent's goal is to find an *optimal policy* π_* that maximizes the *discounted return* $G_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is a *discount factor*. As $\gamma \to 0$, the agent becomes more shortsighted and maximizes the immediate reward. On the other hand, the agent takes the future rewards into account more seriously when $\gamma \to 1$. The agent's task is called *episodic* if $T < \infty$, and such a trajectory is called an *episode*. The *value function* $v_\pi(s)$, also called the state-value function, is the expected return beginning in state s and following policy π afterward. Formally, the value function is defined as $v_\pi(s) = \mathbb{E}_\pi[G_t | s_t = s]$. Likewise, the *q-value function* $q_\pi(s, a)$, also called action-value function, is the expected return beginning in state s, taking action a, and following policy

[1] We follow the notation of Sutton and Barto [161] and use the r_{t+1} instead of r_t to denote that reward and next state s_{t+1} are determined jointly.

π afterward. Formally, the value function is defined as $q_\pi(s,a) = \mathbb{E}_\pi[G_t|s_t = s, a_t = a]$. In the DRL, the Markov property is often relaxed so that the agent does not have to be fully aware of the environmental state. Instead, the agent observes only a partial state known as *observation* o_t (retains far less information compared to the entire state of the environment), forming the Partially Observable MDP (POMDP). One step further is the cooperative Multi-Agent setting, where the agents act in parallel, extending the POMDP to the Decentralized POMDP (dec-POMDP). We refer to Sutton and Barto [161] for further details on the formal background.

4.2 Learning Internal Label Placement

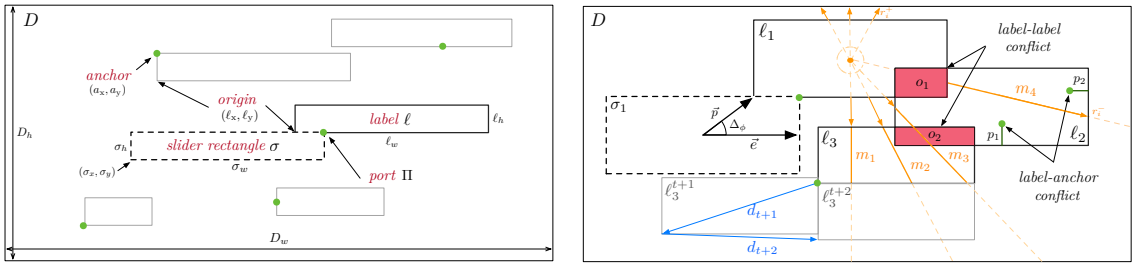
Traditionally, supervised learning methods require a substantial amount of labeled data samples to train the model. However, this requirement poses a significant challenge, as no such labeled dataset currently exists for the point-feature label problem. Creating a dataset for this purpose would require access to an immense number of high-quality drawings with labels and time-consuming annotations. The novel approach presented in this work tackles this problem by posing the point-feature label placement as a reinforcement learning problem, which circumvents the challenge posed by the absence of ground-truth label placement datasets, a crucial requirement for traditional supervised learning methods.

Designing the labeling problem as MARL has several significant benefits over SARL. First, when posing the label placement as SARL, the agent acts as a supervisor managing all labels at once. However, in that case, the observation and action space size changes with a varying number of labels, which contradicts the RL's prerequisite of fixed-sized observation and action spaces. On the contrary, with the abstraction of an agent for each label in MARL, an agent's individual observation and action space can be implicitly designed as fixed-sized. Therefore, the number of agents (*i.e.*, copies of trained strategy) varies in MARL compared to the observation and action space size in SARL. Second, even if one would overcome the variability, the SARL observation space would still be several times larger than in MARL, and the neural network architecture would be more complex and have significantly more parameters, resulting in more challenging training. Third, one would have to collect many more trajectories to train the super agent, as the trajectory in SARL is a collection of the observations, actions, and rewards of all labels. On the contrary, the trajectory of each individual agent can be used to improve the shared strategy in MARL. Therefore, due to the mentioned properties, we decided to represent each label by an agent, which finally unfolds into a Multi-Agent Deep Reinforcement Learning (MADRL) problem. From now on, we also refer to an agent as a label or a label agent interchangeably. Similarly, we refer to a point feature as an *anchor*.

The proposed method is designed explicitly for *adjacent* PFLP, meaning a label can be placed only around its anchor. Our intention is to find a conflict-free label position for each anchor (denoted as *complete* labeling), and if such a position does not exist for all or cannot be found by the method, we call the labeling *incomplete*. The emergence of label placement strategy and rules, in RL referred to as a policy, of our method is driven entirely by the learning process contrary to the existing hand-crafted methods.

4.2.1 Environment

In the following text, we follow the terminology and definitions defined by Bekos *et al.* [20], later extended by Bobák *et al.* [30].



(a) (b)

Figure 4.1: Illustration (a) describes the used terminology and definitions. Illustration (b) depicts the observed modalities and defined action Δ_ϕ. The orange dashed lines represent rays providing the mapping vector of modalities. A ray begins at the label's center, but the actual reading of range starts at the label's bound. We collect the distance and type of the nearest intersection (*i.e.*, label, anchor, or bound) for each ray. The distance is positive (denoted by the plus sign subscript) if the ray hits the label from the outside. On the other hand, the distance is negative if the ray hits the label from the inside; see the ray denoted by r_i^-. Furthermore, we compute the mass of bodies of labels (denoted by the $m_{1\ldots4}$) that the ray went through till the bound of the environment. The overlap modality is depicted by the area of o_1 and o_2. The value for ℓ_1 is just the area of o_1, but $o_1 + o_2$ for ℓ_2 as it is in a label-label conflict with ℓ_2 and ℓ_3. Similarly, we demonstrate the penetration distance for ℓ_2 as a sum of p_1 and p_2. Finally, we show the displacement distance d_{t+1} and d_{t+2} time steps $t+1$ and $t+2$, respectively. The total distance traveled is the cumulative sum of $d_t,\ t \in [0, T]$. The illustration does not show a complete description of the state for the illustration clarity (*e.g.*, only a few rays are visualized, and the labels corresponding to anchors producing the penetration p_1 and p_2 are not shown).

Inspired by the interface of OpenAI Gym [34], we transformed the adjacent PFLP problem into a custom-developed environment, referred to as `AdjacentPFLEnv`, to facilitate the reinforcement learning paradigm. The proposed environment consists of a set of anchors $\mathcal{A}$, each defined by its coordinates $(a_\mathrm{x}, a_\mathrm{y})$ enclosed within rectangular *drawing region* D of dimensions $(D_\mathrm{w}, D_\mathrm{h})$. Each anchor is paired with an axis-aligned box denoted as *label agent* ℓ defined by its *origin* coordinates $(\ell_\mathrm{x}, \ell_\mathrm{y})$ and dimensions $(\ell_\mathrm{w}, \ell_\mathrm{h})$. A set of all label agents within the environment is denoted as $\mathcal{L}$. The label agent's origin coordinates lie on the circumference of the *slider rectangle* σ, whose origin is defined as $(\sigma_x, \sigma_y) = (a_x - \ell_w, a_y - \ell_h)$ and dimensions as $(\sigma_w, \sigma_h) = (\ell_\mathrm{w}, \ell_\mathrm{h})$. Finally, each label agent ℓ remains tethered to its respective anchor via an attachment point denoted as *port* Π; see Figure 4.1(a). We derive the initial origin of an associated label agent as

$$\ell_x = \mathrm{clip}(\mathrm{a_x}, 0, \mathrm{D_w} - \ell_\mathrm{w}), \tag{4.1a}$$

$$\ell_y = \mathrm{clip}(\mathrm{a_y}, 0, \mathrm{D_h} - \ell_\mathrm{h}), \tag{4.1b}$$

where $\mathrm{clip}(x, b_l, b_u)$ is a piecewise function that clips the value x between lower b_l and upper b_u bound. Therefore, a label agent is placed at the initial state s_0 primarily at the most preferred position in the upper right quadrant of the 4-position model, despite the fact that agents can be in *conflict* (*i.e.*, agent-agent overlap, agent-anchor penetration); see Figure 4.1(b).

In the training phase, the anchor coordinates and dimensions of associated label agents are randomized at the initial state s_0. Coordinates of an anchor are drawn from the uniform

distribution such as $a_x \sim U(0, D_w)$ and $a_y \sim U(0, D_h)$. Dimensions of label agents are determined in the similar fashion $\ell_w \sim U(0.1D_w, 0.15D_w)$, and we fixed $\ell_h = 0.05D_h$. The environment is populated only by one to two label agents, none of which or both of which overlap with the other agent, and we chose $D_w = 600$, $D_h = 400$ pixels. The environment terminates at the fixed horizon of $T = 100$ steps. After termination, a new randomized configuration of label agents is populated in the environment and presented to the training algorithm. We apply a fixed-horizon approach to avoid a non-episodic behavior (*i.e.*, infinite horizon) when conflict-free label placement does not exist. Furthermore, a fixed horizon helps stabilize the label agent's position after finding a conflict-free arrangement by allowing the agent to discover that any additional action can lead to a deterioration of the reward.

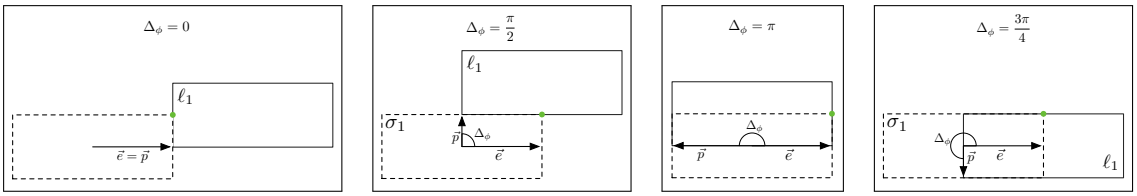
In the evaluation phase, we can populate the environment with any number of label agents, even though we trained the policy with only one to two agents. We are leveraging the capability of RL to generalize to *instances* (*i.e.*, specific configurations of an environment) unseen during the training to find the label placement for hundreds of anchors rather than just two. Moreover, the size of the drawing area, label size, and other mentioned parameters can be selected arbitrarily.

4.2.2 Observation Space

We represent the state of the environment for each label agent ℓ solely by the local same-shaped *observation vector* o_t^ℓ. This approach enables us to leverage all individual trajectories to train a shared policy and facilitate decentralized execution. Furthermore, we solely rely on sensor-based data, raw data acquired directly from the environment, instead of image data. Image data, represented as raster images or pixel matrices, typically demand greater storage and computational resources. By opting for sensor-based data, we effectively reduce the size of the observation vectors. To capture the semantics of the observed modalities, we divided the observation o_t^ℓ into two distinct vectors: the *mapping vector* M and the *self-aware vector* S. The observation vector o_t^ℓ is then obtained by concatenating M and S.

The modalities captured by the mapping vector $M = [\texttt{d}, \texttt{t}, \texttt{c}, \texttt{m}]$ serve to encode the agent's surroundings through the use of 32 ray sensors, which are uniformly distributed around the label boundaries and function in a similar manner to LiDAR sensors. These sensors measure the distance `d` to the closest intersection point, as well as the type `t` of an object that the ray intersects (*i.e.*, label, anchor, bounds of the environment). Additionally, the mapping vector captures the count `c` and mass `m` of the labels that the ray passed through.

The self-aware vector $S = [\texttt{O}, \texttt{D}, \texttt{Ape}, \texttt{Apr}, \texttt{Ad}, \texttt{T}]$ supplies local modalities that pertain mainly to the agent's conflicts, including overlaps with other agents and penetrations with anchors. We define the overlap modality `O` as a sum of the occluded area between the given agent and the other agents being in conflict. The displacement `D` represents the Euclidean distance of the agent's origins between two consecutive time steps t and $t+1$. Furthermore, we define penetration distance as the Euclidean distance between the penetrated anchor and the nearest point of escape on the circumference of the label. Like the overlap modality, we define the penetration modality `Ape` as a sum of the penetration distances between a given agent and its anchor or the other agents' anchors being in conflict. For both modalities, we also provide the agent with a count of conflicts relative to the total number of label agents. Additionally, the agent observes the Euclidean distance to its anchor `Ad`, the relative position



(a) (b) (c) (d)

Figure 4.2: A detailed illustration of the agent's action Δ_ϕ. For $\Delta_\phi : 0 \rightarrow 2\pi$, the origin of the label $[\ell_x,\ \ell_y]$ moves counterclockwise along the circumference of a slider rectangle σ (denoted by a dashed line), whose origin is defined as $(\sigma_x, \sigma_y) = (a_x - \ell_w, a_y - \ell_h)$ and dimensions as $(\sigma_w, \sigma_h) = (\ell_\mathrm{w}, \ell_\mathrm{h})$.

of the anchor to the label agent's port `Apr`, and finally, the information about elapsed time steps `T`.

We normalize all observation modalities. An overview of the observed modalities is presented in Figure 4.1(b), while a comprehensive description is available in Appendix B[2].

4.2.3 Action Space

We designed the action space of agents such that they can change the origin continuously without discretization to the environment's raster, allowing subpixel precision and independence of the drawing area dimensions. Therefore, we define a continuous action Δ_ϕ representing the angle between basis vector $\vec{e} = (1, 0)$ and vector $\vec{p} = (p_x, p_y)$, where point $[p_x, p_y] \equiv [\ell_x,\ \ell_y]$ belongs to the circumference of the slider rectangle for a given label ℓ; see Figure 4.1(b) and Figure 4.2. We use one-dimensional action space Δ_ϕ rather than an intuitive two-dimensional action space of Δ_x and Δ_y to simplify the learning process. In our experience, multi-dimensional action can introduce unnecessary coordination complexity. Given $\Delta_\phi \in (0, 2\pi)$, we define the origin of a label in quadrant Q1 as

$$[\ell_x,\ \ell_y]_{t+1} = \begin{cases} \frac{\ell_w}{2}[1,\ \tan(\Delta_\phi)] & 0 \leq \Delta_\phi < \arctan\left(\frac{\ell_h}{\ell_w}\right) \\ \frac{\ell_h}{2}[\cot(\Delta_\phi),\ 1] & \arctan\left(\frac{\ell_h}{\ell_w}\right) \leq \Delta_\phi < \frac{\pi}{2}. \end{cases} \tag{4.2}$$

The derivation for quadrants Q2 to Q4 is trivial – the idea is to transform the Δ_ϕ into the Q1 and then flip the position to the corresponding quadrant. For clarity, we omitted use of $\Delta_\phi \in [-1, 1]$ in our implementation; nevertheless, the previous declaration holds.

To facilitate continuous action space with an infinite number of actions, we define a policy as a parameterized Gaussian distribution. Therefore, instead of learning the probabilities of all possible actions, which is infeasible, we learn statistics of the distribution. Formally, we define policy as

$$\pi_\theta(\Delta_\phi | s_t) = \mathcal{N}\left(\mu_\theta(s_t), \sigma^2_\theta(s_t)\right), \tag{4.3}$$

where s_t is the current state at time step t, $\mu_\theta(s_t)$ and $\sigma^2_\theta(s_t)$ is the mean and variance of the distribution parametrized by the neural network parameters $\theta \in \mathbb{R}^d$ further described in Section 4.2.5.

[2]Supplementary material related to this project can be also accessed via project page at `http://cphoto.fit.vutbr.cz/reinforced-labels`.

4.2.4 Reward

Our goal in this work involves the emergence of cooperation among agents to find a conflict-free label position for each anchor. Therefore, we combine two types of rewards as suggested by Nguyen *et al.* [131]. The *local* reward r^{local} assigns an agent its feedback solely based on individual efforts, while the *global* reward r^{global} gives an agent its feedback based on the entire state of the environment. As a local reward, the agent receives step-wise penalization (*i.e.*, negative reward) for being in label-label conflict with another agent, meaning both label agents overlap with each other. We compute the *overlap value* $o(\ell)$ in the same way as the overlap modality, as a sum of the occluded area between a given agent and the other agents in conflict. The global reward is the composition of local rewards among individual agents. Formally, we define the local reward r^{local} and global reward r^{global} as

$$r_{t+1}^{\text{local}}(\ell) = -o(\ell) \qquad\qquad r_{t+1}^{\text{global}} = \sum_{\ell \in \mathcal{L}} r_{t+1}^{\text{local}}(\ell), \tag{4.4}$$

where t is the current time step, $o(\ell)$ is the overlap value. Based on these definitions, we define the total reward for a label agent ℓ as

$$r_{t+1}^{\text{total}}(\ell) = (1 - w) \cdot r_{t+1}^{\text{global}} + w \cdot r_{t+1}^{\text{local}}(\ell). \tag{4.5}$$

By observing the influence of various weight values on the final reward of the trained policy, we recommend using $w = 0.5$.

4.2.5 Policy & Value Network Architecture

We designed an efficient yet straightforward feedforward network with just less than half of a million (412 thousand) parameters, as depicted in Figure 4.3. The architecture consists of two input heads – *mapping* and *self-aware*, and two output branches – *value* and *policy*. The ray observations from sensors constituting the mapping head are first passed through a circular 1D-convolution layer proposed by Schubert *et al.* [152] to capture the correlation between individual readings, including the borders of the tensor. The intermediate representation is reshaped to form a 1D tensor. The other observation modalities forming the self-aware head are concatenated and embedded by a dense layer. The mapping and self-aware heads are concatenated and passed by a final shared dense layer. At the end of the architecture, two separate dense layers split the outcome into the value and policy branches.

4.2.6 Training

We adopt a *parameter sharing* in which each agent utilizes the same policy network with identical parameters. This approach allows us to optimize the parameters of the proposed network using the trajectories of the individual agents. We update the parameters of the policy network in a *policy gradient* fashion – meaning the parameters are updated based on the gradient of an estimate of expected return with respect to the policy parameters. In particular, we utilize Proximal Policy Optimization (PrPO), one of the most prominent actor-critic policy gradient methods proposed by Schulman *et al.* [154]. The method is best known for its relative simplicity while preserving the convergence properties of more complex predecessors. We refer to the survey of Arulkumaran *et al.* [8] for further details on the DRL algorithms. The goal is to optimize the parameters θ that maximize the expected

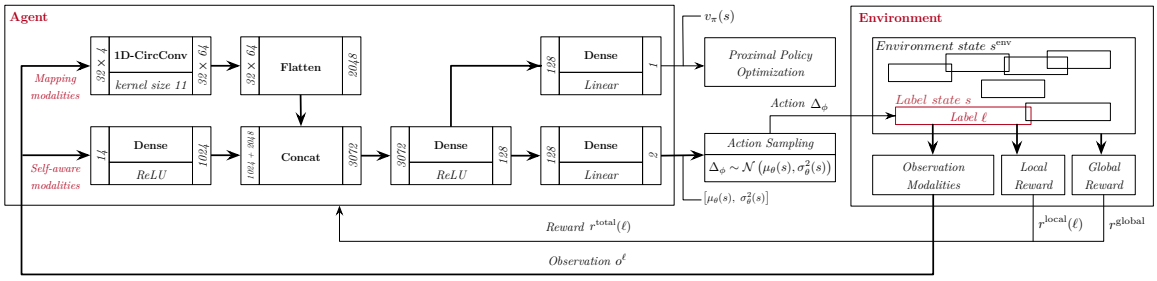


Figure 4.3: Overview of Reinforced Labels and its neural network architecture. The proposed `AdjacentPFLEnv` environment describes the adjacent point-feature labeling problem. We are proposing a Multi-Agent approach, where an agent controls the origin of a label. We use parameter sharing, meaning each agent acts according to the same policy network with identical parameters. Therefore, we can use trajectories of individual agents to optimize the parameters of the proposed network. We utilize only local agent observations o^ℓ, enabling parallel execution in the evaluation phase. We provide the agent feedback by local rewards r^{local} and global rewards r^{global} only during the training; the reward is not used in the evaluation phase. The mapping and self-aware modalities are passed through the network, resulting in a state value $v(s)$ and parameters of normal distribution $\mu_\theta(s)$ and $\sigma^2_\theta(s)$. The training algorithm [154] uses the $v(s)$ to optimize the parameters of the network. The distribution parameters are used to sample an action Δ_ϕ. Finally, the agent acts according to the action, which might translate (*i.e.*, the $\Delta_\phi \neq 0$) into the change of the label's origin.

discounted return $\mathbb{E}[G_t]$. The surrogate objective of PrPO is defined as

$$L(\theta)^{CLIP} = \mathbb{E}\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right], \tag{4.6}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ is the probability ratio of the new and old policies. Value of $\hat{A}_t$ is the estimator of the *advantage function*, computed by Generic Advantage Estimation introduced by Schulman *et al.* [153], describing whether choosing action a_t in state s_t is better or worse than the average action of policy π. One of the essential concepts of PrPO is the clip operation on top of the probability ratio $r_t(\theta)$ that discourages the policy from dramatically changing between training iterations, resulting in convergence issues. Lower values of ϵ correspond to more consistent policy improvements. On the other hand, higher values yield greater variance and volatility of convergence. The value network objective is formulated as regression via mean square error as

$$L^{VF}(\theta) = \mathbb{E}\left[(v_\pi(s_t|\theta) - v^{\text{target}})^2\right], \tag{4.7}$$

where $v^{\text{target}} = r_t + \gamma r_{t+1} + \ldots + \gamma^{T-t} v_{\pi_{\theta old}}(s_t)$. Because the proposed neural network architecture shares parameters between policy and value networks, we combine the aforementioned objectives to a composed loss to train both networks simultaneously as $L(\theta) = L(\theta)^{CLIP} - L^{VF}(\theta)$. A detailed list of hyperparameters is available in Appendix B.

4.2.7 Inference

Once the policy network is trained, we can use it to evaluate an instance with an arbitrary number of anchors, size of labels, and dimensions of the drawing area, all due to the

proposed MADRL design properties and choice of the observed modalities and actions. At the beginning of the evaluation, we initialize the environment according to Equation 4.1. The evaluation step for an agent ℓ consists of a collection of observation modalities to form vector o_t^ℓ and passing it through the shared policy network, resulting in action Δ_ϕ. This sequence produces a new environment state s_{t+1} and a new observation vector o_{t+1}^ℓ. The process is repeated for all agents until no conflicts are present or the horizon T is hit.

4.3 Ablation Study

We conducted an ablation study to fine-tune the proposed method and validate our design choices. The goal is to shed light on the impact of each component within our method and to understand how variations in these components can influence overall performance. Additionally, the findings from these ablations open up the potential for future research pathways, creating opportunities for further optimization and refinement of our technique. To measure the performance of our method under different ablations, we introduce a novel metric and benchmark dataset.

4.3.1 Completeness Metric

Labeling metrics typically measure the quantity of non-conflicting labels. Such metrics, however, are best suited for methods that ensure conflict-free label layouts. Conversely, methods not offering such assurances may calculate performance based on the sum of overlapping regions. Yet, from the perspective of the established label placement rules [82,185], any overlap, whether slight or significant, deems a layout non-conflict-free. Given this landscape, we observed a gap: while conflict-free methods may omit labels to avoid conflicts, it is uncertain how the overlap regions would appear. Conversely, methods without a conflict-free guarantee do not indicate labels that should be discarded to achieve a conflict-free layout. To close the gap, we introduce the *completeness* metric. Our work focuses on finding a conflict-free label position for each anchor. If such a position does not exist for all anchors or cannot be found by the method, we call the labeling *incomplete*. On the other hand, we denote the labeling as *complete* if all anchors are annotated without conflict.

We measure the performance of label placement methods by the completeness metric representing the percentage of complete labelings for a given set of instances. For example, let D be a dataset of 10 instances. Let M_1 be PFLP method that found eight complete conflict-free layouts without the need of removing any label and two incomplete conflict-free layouts with several removed labels. Let M_2 be a method with the same PFLP properties as M_1 that found nine complete conflict-free layouts and one incomplete layout with remaining conflicts (*i.e.*, label-label or label-anchor conflict). The completeness of M_1 is 80%, and the completeness of M_2 is 90%, as the latter method found more complete layouts out of a set of ten given instances. Therefore, method M_2 performs better than method M_1.

4.3.2 Benchmark Dataset

We have created a benchmark dataset to compare outcomes and performance among the evaluated label placement methods. We split the dataset into two parts – *compact* and *volume* datasets. To endorse the standardized evaluation of labeling methods, we provide the benchmark dataset as supplementary material.

Model	*Completeness* %	*Steps* -	*Inference* ms	*Parameters* millions
RFL_Conv	**97.9**	**30.721**	**3.077**	**0.412**
RFL_ConvLn	96.8	37.333	3.513	0.412
RFL_2Dns	**96.4**	**41.466**	**2.775**	**0.673**
RFL_2DnsLn	94.5	59.353	2.998	0.673
RFL_1Dns	**95.4**	**53.081**	**2.665**	**0.833**
RFL_1DnsLn	94.7	63.307	2.882	0.833
RFL_rllib	90.6	81.905	2.008	0.206

Table 4.1: Ablations of neural network architecture evaluated on the compact dataset. We computed the overall completeness metric for the dataset to summarize the performance and simplify the comparison of the ablations examined. Furthermore, we provide the average number of elapsed steps needed to solve instances in the dataset, inference time per step, and the number of parameters to illustrate the complexity of the neural network. The notation is explained in Section 4.3.3.

We generated the anchor coordinates using a pseudo-random number generator with uniform distribution common across multiple bodies of previous work [107, 138, 183]. In the compact dataset, we sampled the anchor coordinates from an area of 600×400 and sequentially increased the number of drawn samples by five, ranging from 5 to 50. For the volume dataset, we drew coordinates from an area of 2400×1600 and consecutively raised their count by 50, going from 100 to 600. For each number of anchors, we generated ten instances. Therefore, the entire dataset consists of 41 250 anchors divided into 210 instances. Furthermore, we randomly formed the corresponding labels so that the text consists of three to seven capital letters from the English alphabet.

It is worth noting that the benchmark dataset also contains instances that cannot be solved, as it was created using a pseudo-random number generator, due to the factual inexistence of a conflict-free layout. In such a case, any labeling method cannot produce complete labeling. However, such instances do not influence the comparison of labeling methods on the proposed dataset because these instances always affect all methods the same.

4.3.3 Architecture Ablations

In order to justify the design choices behind our proposed architecture, we performed several architectural ablations. Each ablation was evaluated on the compact dataset ten times to adequately account for the inherent stochastic properties of Reinforced Labels (RFL). In an effort to summarize the performance and facilitate the comparison of the examined ablations, we report the overall completeness metric as a single average value for the compact dataset. The results of these evaluations can be found in Table 4.1.

We categorized the ablations into four groups, starting with the proposed architecture and systematically removing/varying its components. Furthermore, we also evaluated the universal baseline architecture of a leading reinforcement learning library RLLib utilized in our development [108]. The first ablation group **RFL_Conv*** contains a custom two-head, two-branch convolution-based architecture as described in Section 4.2.5. The mapping modalities are first concatenated into vector M, embedded by a circular 1D-convolution

layer, and optionally passed through a layer normalization. The self-aware modalities are concatenated into vector S, embedded by a dense layer, and optionally passed through a layer normalization denoted by **Ln** suffix. Ultimately, the outcomes are concatenated, and two separate dense layers split the outcome into the value and policy branches. The second ablation group **RFL_2Dns*** retains the two-head, two-branch architecture but excludes the circular 1D-convolution layer. Instead, the mapping modalities are embedded by a dense layer. The third ablation group **RFL_1Dns*** includes custom single-head, two-branch architecture. The observed modalities M and S are merged, embedded by two dense layers, and optionally passed through a layer normalization. At the end of the architecture, two separate dense layers split the outcome into the value and policy branches. The final ablation **RFL_rllib** holds the default baseline architecture of RLLib. The observed modalities M and S are concatenated and separated into the value and policy by two triple-dense-layer branches.

Table 4.1 shows the results of the architectural ablations. The best-performing model is the RFL_Conv, as described in Section 4.2.5, achieving an overall completeness of 97.9%. We emphasize the importance of the circular 1D-convolution layer as, in addition to classical convolution, the circular one captures the correlation between the borders of the tensor. The second-best score was achieved by RFL_2Dns, without circular convolution but preserving the two-head scheme, achieving 96.4%. However, the minor decline in completeness is accompanied by a significant increase in the average number of steps needed to solve instances in the dataset. Changing the architecture to a single-head scheme (RFL_1Dns) leads to a further decrease in performance to the completeness of 95.4%. We also evaluated the default universal model RFL_rllib, which performs the worst, achieving significantly lower completeness of 90.6%. The main difference between the architecture of RFL_rllib and the other variants is the absence of a shared dense layer before splitting the outcomes into the value and policy branches. Therefore, we argue that the presence of a shared dense layer is a vital part of the proposed architecture as it carries the most significant difference in completeness. Finally, the results of our ablation study consistently show that layer normalization is an inappropriate architecture component for the given label-related modalities, always resulting in an overall performance drop. In the following text, we simplify our notation and use RFL to denote the RFL_Conv model.

4.3.4 Observation Ablations

Similarly to the previous section, we ablated the observed modalities to justify our design choices of representing the agent's state. We start with the proposed observation set and then systematically remove/vary the included modalities fed into the best-performing architecture RFL_Conv. As described in Section 4.2.2, the proposed vector of observed modalities o_t consists of mapping modalities denoted by **M** and self-aware modalities denoted by **S**. Ray-based mapping modalities include distance to the nearest intersection **d**, type of the nearest intersected object **t** (i.e., label, anchor, bounds of the environment), and count **c** and mass **m** of labels that the ray went through. Self-aware observations include overlap **O**, displacement **D**, anchor penetration distance **Ape**, anchor-port distance **Apr**, anchor-origin distance **Ad**, and time step **T** modalities. To investigate the importance of ray-based modalities, we also include mapping modalities based on the agent's origin **Or** and size **Si** as a replacement of ray casting.

The results are presented in Table 4.2. For clarity, we provide a numerical notation of each set of evaluated modalities (*i.e.*, numbers in the first column of the table). First

Set	Modalities	*Completeness* %	*Steps* -	*Inference* ms	*Observation* ms
1	`M008[d cm]S[O ApeAprAdT]`	96.8	36.791	2.723	19.870
2	`M016[d cm]S[O ApeAprAdT]`	96.8	41.389	2.713	23.444
3	**`M032[d cm]S[O ApeAprAdT]`**	**97.9**	**30.721**	**2.883**	**30.645**
4	`M064[d cm]S[O ApeAprAdT]`	97.9	32.159	3.077	40.400
5	`M128[d cm]S[O ApeAprAdT]`	97.7	36.630	3.150	63.260
6	**`M[d cm]S[O ApeAprAdT]`**	**97.9**	**30.721**	**3.077**	**30.645**
7	`M[OrSi]S[O ApeAprAdT]`	96.8	46.382	2.679	16.013
8	`M[d cm]S[  ApeAprAdT]`	45.3	298.759	1.999	27.096
9	`M[d cm]S[O    AprAdT]`	90.3	75.955	2.506	29.012
10	**`M[d cm]S[O Ape   AdT]`**	**97.9**	**33.594**	**2.575**	**26.721**
11	`M[d cm]S[O ApeApr  T]`	97.4	34.255	2.698	28.690
12	`M[d cm]S[O ApeAprAd ]`	97.3	33.095	2.722	28.606
13	`M[  cm]S[O ApeAprAdT]`	96.7	36.610	2.569	27.742
14	`M[d  m]S[O ApeAprAdT]`	97.0	41.403	2.655	30.215
15	`M[d c ]S[O ApeAprAdT]`	97.2	34.945	2.752	28.332
16	`M[dtcm]S[ODApeAprAdT]`	97.2	35.352	2.740	29.763
17	`M[OrSi]S[ODApeAprAdT]`	94.8	61.165	2.539	15.829
18	`M[dtcm]S[ DApeAprAdT]`	50.6	270.392	2.007	41.181
19	`M[dtcm]S[O ApeAprAdT]`	97.5	40.649	2.758	28.917
20	`M[dtcm]S[OD   AprAdT]`	86.4	98.084	2.567	28.735
21	`M[dtcm]S[ODApe   AdT]`	96.8	46.255	2.681	27.909
22	`M[dtcm]S[ODApeApr  T]`	93.9	66.275	2.578	28.057
23	`M[dtcm]S[ODApeAprAd ]`	95.1	55.721	2.609	27.961
24	`M[ tcm]S[ODApeAprAdT]`	95.2	59.602	2.535	27.942
25	**`M[d cm]S[ODApeAprAdT]`**	**97.6**	**41.915**	**2.912**	**30.325**
26	`M[dt m]S[ODApeAprAdT]`	96.4	54.344	2.579	28.562
27	`M[dtc ]S[ODApeAprAdT]`	96.8	39.204	2.615	28.272
28	**`M[dtcm]S[O    ]`**	**86.0**	**94.781**	**2.628**	**27.442**
29	`M[OrSi]S[O    ]`	83.3	96.557	2.522	14.972
30	`M[    ]S[O    ]`	82.4	101.750	2.414	14.525

Table 4.2: Ablations of observed modalities evaluated on the compact dataset. We provide overall completeness, the average number of elapsed steps needed to solve the dataset, inference time per step, and observation creation time per step. The notation is explained in Section 4.3.4.

and foremost, the results show that removing the overlap modality (comparing sets 18 with 16 and 8 with 6) dramatically degrades the performance and highly impacts the average number of steps needed to solve the labeling. Therefore, we argue that the overlap modality is the most crucial modality that effectively allows an agent to avoid label-label conflicts. Similarly, removing the anchor penetration modality `Ape` (comparing sets 20 with 16 and 9 with 6) leads to a significant performance decline and moderately impacts the average number of steps needed to solve the labeling. We argue that the penetration modality effectively allows an agent to avoid label-anchor conflicts. Replacing the mapping ray-based modalities with modalities based on the agent's origin `Or` and size `Si` (comparing sets 17 with 16) advocates the presence of ray casting in our method. Ray-based modalities deliver (a) higher completeness and (b) effectively cut the average number of needed steps nearly to half. Interestingly, further removal of displacement modality `D` (comparing set 19 with 16) positively impacts the performance. We argue the displacement modality

(cumulative distance traveled) decreases the number of steps needed for an expense of lower completeness. A similar explanation applies to ray-type modality (comparing set 25 with 16). Therefore, we omitted displacement and ray-type modality due to the negative impact on completeness and conducted a second round of ablations (set 15 to 6). The results show an additional increase in completeness and a decline in the number of steps needed, please compare set 16 with 6. Further findings remain consistent with the first round of ablations (set 27 to 16) except for omitting the proximity modality (set 10) that achieves the same performance, but a higher number of steps is needed in comparison with set 6.

Finally, we experimented with a number of rays, as shown by sets 1 through 5, where **the number following `M`** denotes the number of casted rays. The results indicate that the best option is to cast 32 and 64 rays to achieve the best completeness. However, casting 64 rays demands more resources, leading to increased observation creation time compared to 32 rays. Reducing as well as increasing the number of rays leads to lower completeness. We hypothesize that the information is likely too sparse at the lower end, not providing sufficient detail. Conversely, opting for 128 over 64 rays might not offer substantial additional information and, combined with an undersized convolution filter, lead to a slight decrease in performance. Nevertheless, the confirmation of our hypothesis remains open for future research. In the following text, we simplify our notation and use `RFL` to denote the `RFL_Conv` model combined with modalities of set 3.

4.4 Comparison with State of the Art

We employ the completeness metric and benchmark dataset, as outlined in Section 4.3.1 and Section 4.3.2, to compare our Reinforced Labels (`RFL`) with several published methods, beginning with the implementation of Particle-Based Labeling (`PBL`) proposed by Luboschik *et al.* [115]. The latter method attempts to position each label sequentially, first with the fixed 4- and 8-position model, then using the slider model, and finally with a spiral-based distant model. Furthermore, the method is greedy, and such cannot change the position of a label after it has been placed. Moreover, labels that the method cannot position without a conflict are removed from the calculated label layout. Therefore, the approach may produce an incomplete label layout. We separate `PBL` into two variants. The first, denoted as `PBL-A`, involves only the fixed 4-, 8-position, and slider models resulting in **adjacent** label placement. The other variant, referred to as `PBL-AD`, applies the spiral-based distant model in addition to the fixed 4-, 8-position, and slider models, resulting in a combination of both **adjacent and distant** label placement (*i.e.*, a label can be placed farther away from its anchor, and a leader line maintains the correspondence). Please be aware that `PBL-AD` cannot be directly compared with adjacent-only methods, such as `RFL`, given the fact that the distant labels offer a greater degree of freedom. As a result, we expect `PBL-AD` to achieve a higher completeness score than adjacent-only methods. Nevertheless, we include the `PBL-AD` in the evaluation to compare the `RFL` with a method that uses distant labels. Additionally, we compare `RFL` to the Rapid Labels (`RAPL`) [138], a GPU-accelerated greedy and adjacent-only method that leverages the 8-position model. Similarly to `PBL`, `RAPL` may produce an incomplete layout as labels that cannot be placed without a conflict by the method are removed from the calculated layout. Table 4.3 presents a comparison of the evaluated label placement methods. Finally, we introduce an untrained version of `RFL` with randomly initialized weights (abbreviated as `RFL-random`) to validate that the `RFL` learns a reasonable policy.

	RFL	RAPL	PBL-AD	PBL-A
Paradigm	Machine Learning	Algorithm	Algorithm	Algorithm
Position model	Slider	Fixed	Fixed/Slider	Fixed/Slider
Problem definition	Reward	Value based on rules	Rules	Rules
Label position	Continuous	Discrete	Discrete	Discrete
Heuristic	Non-greedy	Greedy	Greedy	Greedy
Computation time	Non-interactive	Highly interactive	Interactive (when using collision particles)	Interactive (when using collision particles)
Completeness	High	Low	High (achieved by distant labels)	Low
Resolution-independent	Yes	No	Yes	Yes

Table 4.3: Comparison of the evaluated label placement methods. The desirable properties are depicted in green, and less desirable properties are marked in orange and red. The computation time requirements are contingent upon the specific use case. For scenarios where labeling needs to be pre-computed with a high level of completeness, such as cartographic maps, technical drawings, or medical atlases, the RFL method is an appropriate choice. Conversely, in interactive applications such as games or viewers, the trade-off of decreased labeling completeness may be acceptable to ensure faster computation times. Please note that the *Position model* for the PBL-AD and PBL-A methods is *Fixed/Slider* as these methods first try to position labels on the fixed positions, and only if that is not possible, they try several positions between the fixed ones.

4.4.1 Quantitative Results

We trained the RFL policy on randomly generated instances previously described in Section 4.2.1 with at the most two anchors, therefore with up to two agents within the environment, for an hour on a computation node equipped with 2x AMD EPYC™ 7H12, 64-core, 2.6 GHz CPUs without a GPU accelerator. To optimize the parameters of the neural network, we utilize PrPO implementation from the RLLib framework [108]. Specifically, we used 119 cores for rollout workers to collect agents' experiences (*i.e.*, observations and rewards) and a single core for the trainer worker responsible for updating the parameters of the proposed network. A detailed description of our training setup is available in Appendix B. We executed the following evaluations on Intel® Core i7-9700K 8-core, 3.60GHz CPU, and NVIDIA GeForce GTX 1660 Ti. To capture the stochastic nature of RFL (see the definition of Equation 4.3), we evaluated the RFL and RFL-random ten times over the benchmark dataset. The other compared methods, PBL-A, PBL-AD, and RAPL, are deterministic, and as such, we evaluated each only once. Therefore, we provide quartiles Q1 and Q2, mean and median statistics for the completeness metric of RFL. Furthermore, we fixate the episode horizon of RFL and RFL-random in the evaluation phase at $T = 500$ steps (recall we set the $T = 100$ for the training).

Figure 4.4(c) and Figure 4.4(d) show the completeness of the compared methods on the compact and volume datasets. Both charts reveal a similar trend. As the number of anchors rises, as does the environment *occupancy* (*i.e.*, a ratio of the total area of the labels to the overall area of the drawing), the completeness of PBL-A, RAPL, and RFL-random decreases rapidly. The reason behind the difference in completeness between 50 anchors in the compact and 100 anchors in the volume dataset is the varying occupancy. It is easier to find complete labeling within a larger space of volume dataset. Besides PBL-AD, which has the advantage of distant labels, the RFL achieves the highest average completeness of 89% and 64% for 50 and 600 anchors, respectively. The random policy of an untrained agent RFL-random corresponding with a chance performs the worst. This fact confirms that the RFL learns a meaningful policy, and its performance is not the outcome of the randomized search. The second worst method in terms of completeness is PBL-A, followed by RAPL. Starting from 300 anchors, PBL-A fails to place all the labels for any instance, resulting

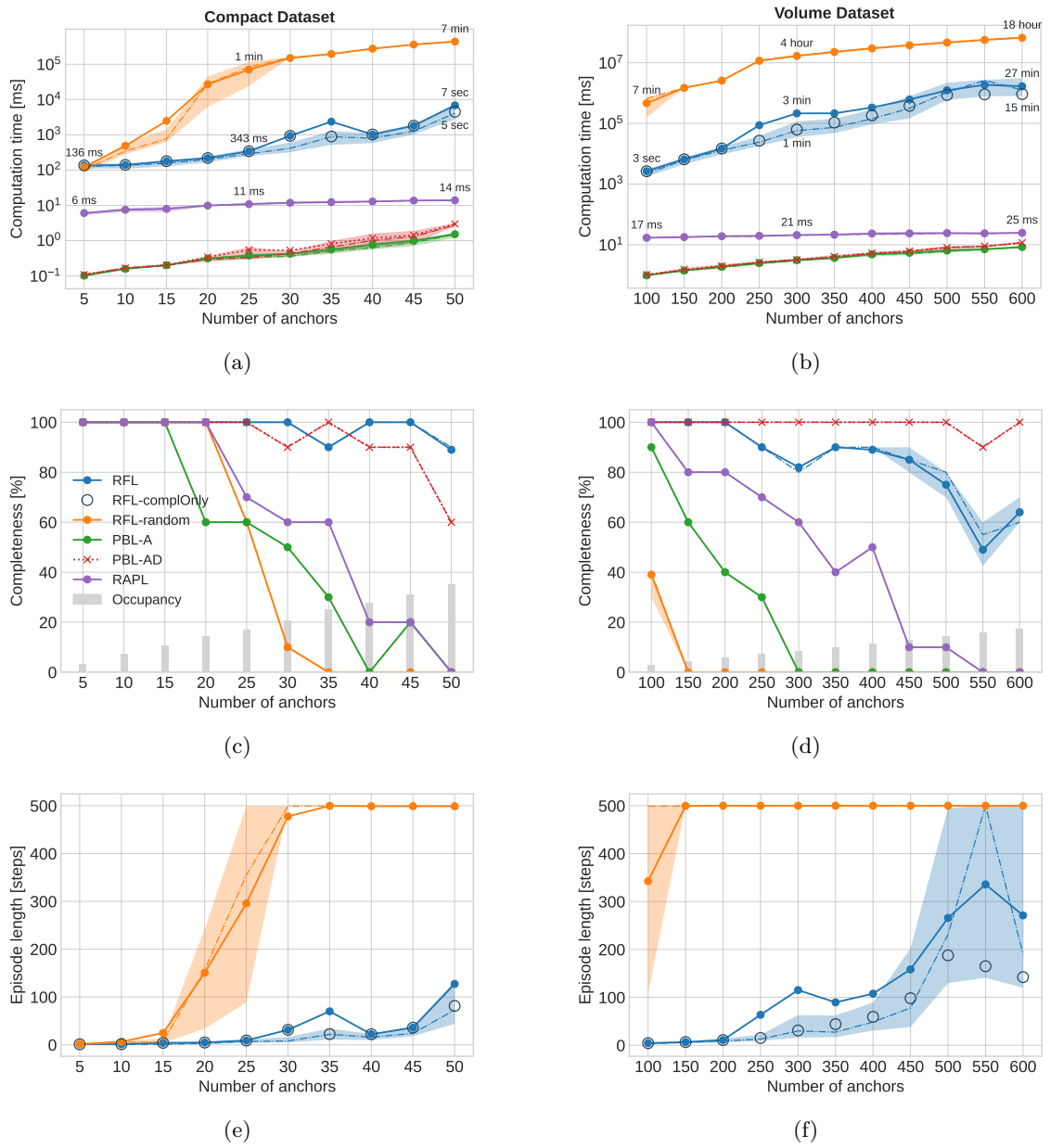


Figure 4.4: Comparison of the examined methods evaluated on the benchmark dataset. Charts (a) and (b) show computation time. Charts (c) and (d) illustrate the completeness on compact and volume datasets. Charts (e) and (f) depict the length of the episode. We recall that `PBL-AD` combining both **adjacent and distant** models cannot be directly compared with **adjacent**-only methods such as `RFL`, `PBL-A`, or `RAPL`.The former `PBL-AD` provides a greater degree of freedom by placing a label further away from its anchor while the additional leader line maintains the correspondence. We acquired the values for `RFL` and `RFL-random` by executing these methods ten times over a set of instances with the same number of anchors. The solid line represents the mean, the shaded area depicts quartiles Q1 and Q2, and the dash-dot line describes the median. The empty circle symbol represents the mean value of complete-only instances found by `RFL` (incomplete instances reaching the fixed horizon are filtered out). In simpler terms, the value conveyed by the symbol provides an answer to the question: "If and only if the `RFL` can solve the given instances, how long does it take on average?"

in 0% completeness. Similarly, at 50 and 550 anchors, PBL-A, RAPL, and RFL-random all achieve 0% completeness.

We also compared RFL with PBL-AD, which combines **adjacent and distant** labels and, therefore, has a higher degree of freedom than the strict slider model of our method. However, it is essential to recognize that the inherent flexibility of PBL-AD, stemming from its ability to position distant labels away from the anchors and the use of leader lines for connection, makes it directly incomparable with adjacent-only methods such as RFL. Moreover, the leader lines that grant PBL-AD its flexibility can also intersect with other labels, anchors, or leader lines, creating potential conflicts. Such conflicts not only diminish the labels' readability but also complicate the association between labels and anchors. Imhof [82] underscores the importance of both label readability and straightforward label-to-anchor association in label layout. Hence, while PBL-AD might seem advantageous regarding placement flexibility, it should be approached cautiously, especially when label layout quality is essential. To our surprise, the PBL-AD performs slightly worse on the compact dataset than RFL and surpasses RFL only at the point of 35 anchors. We further investigated this case and found that the dip in RFL performance is caused by the factual inexistence of a complete conflict-free layout. Therefore, the proposed method outperforms both hand-crafted algorithms RAPL and PBL-A/-AD on the compact dataset. On the volume dataset, the PBL-AD shows stable completeness of 100% through the dataset, except at 550 anchors, where the completeness dips slightly to 90%. We attribute the superior performance of PBL-AD on the volume dataset to the fact that anchors are not spread as evenly across the entire space as in the compact dataset. As a result, PBL-AD can utilize more distant labels for anchors in dense clusters and position them in less dense areas.

Figure 4.4(a) and Figure 4.4(b) depict the dependence of the computation time on the number of anchors. We group all instances with a given number of anchors and compute the aggregated statistics. Therefore, we report quartiles Q1 and Q2, mean and median statistics. The PBL-A and PBL-AD are the fastest methods over the entire benchmark dataset. The RAPL follows with a difference of an order of magnitude that steadily decreases towards 600 anchors. This fact goes along with the authors' statement that the performance gain comes with a more significant number of anchors due to the computation of the Summed Area Table [138]. The second slowest method is RFL. We attribute this to the RLLib's internal inefficiencies (*i.e.*, policies among agents within an environment cannot be evaluated in parallel) and the utilized single-thread ray casting implementation from the Box2D framework[3]. In fact, the observation computation and collection take on average 46% (29 ms) of the computation time per step, of which 2/3 makes up the ray-casting operation. The inference of the proposed architecture carries only about 7% (4 ms) of the time. The final 47% (30 ms) is dissolved in preprocessing observations and actions within the RLLib framework. The slowest method overall was RFL-random, hitting the upper bound of 500 steps. Again, this fact confirms that RFL learns a reasonable policy, and its performance is not an outcome of the randomized search.

Figure 4.4(e) and Figure 4.4(f) illustrate the dependence of RFL and RFL-random on episode length and the number of anchors. The results significantly distinguish the policy of trained and untrained agents corresponding with a chance that a random set of actions reach the complete conflict-free labeling. The difference is mainly visible at 30 and 150 anchors – the policy of RFL-random skyrockets to the horizon of 500 steps while RFL tops 32 and 7 steps at the same point, respectively.

[3]The framework Box2D is available at https://box2d.org.

4.4.2 User Study

Along with the quantitative evaluation, we conducted a user study to determine the preferred methods among users. We assessed the `RFL` method, as well as the state-of-the-art methods `RAPL`, `PBL-A`, and `PBL-AD`, which were also evaluated in the quantitative analysis. To carry out the user study, we utilized instances from the compact dataset containing 30, 40, and 50 anchors and labeled each instance with all the evaluated methods. The labeled instances can be found in Appendix B. We opted for these instances to assess each method with increasing occupancy while avoiding overwhelming the participants when comparing the layouts.

We have designed the user study based on the psychophysical technique of paired comparisons [166]. Specifically, we utilized the two-alternative forced choice (2AFC) paradigm. Each participant was sequentially presented with all possible label layout pairs where, in each pair, the label layouts of the same instance were used and created with different methods. The participants' task was to choose their preferred label layout for each pair in the sequence. To mitigate the learning effect and fatigue, we randomized both the order of the pairs in the sequence and the positions of label layouts (left or right) in pairs. The user study was conducted with 21 participants, consisting of 19 males and two females, with an average age of 21.53 years (ranging from 21 to 24). The average experiment completion time was 6 minutes and 44 seconds, with participants taking anywhere from 2 minutes and 54 seconds to 12 minutes and 32 seconds. Two of the 21 participants were removed using the outlier analysis tool from Pérez-Ortiz and Mantiuk [139], as their results deviated significantly from the others.

We stored the choices in the count matrix $\mathbf{C}$ for each participant individually. Each element c_{ij} in the matrix indicates the number of times that method i was selected over method j. We transformed the per-participant-count matrices $\mathbf{C}$ into a quality score (z-score) scale and calculated statistical significance using a customized MATLAB framework [139]. To transform the matrix $\mathbf{C}$ to the quality score scale, we used Thurstone's Law of Comparative Judgment model concerning Case V [139, 166]. We employed a Two-tailed test with a significance level of $\alpha = 0.05$ to reject the null hypothesis, "there is no clear user preference among the tested methods."

Figure 4.5 presents the quality scores and the statistical significance of the evaluated methods. The results show that the null hypothesis is clearly rejected as the proposed `RFL` method exhibits the best quality score that is significantly better than the quality scores of the remaining evaluated methods. In other words, the proposed `RFL` method was preferred by the users over the remaining evaluated methods. The results also suggest that there is not a significant difference in user preference for the remaining `RAPL`, `PBL-A`, and `PBL-AD` methods.

4.4.3 Discussion

The outcome of the comparison is manifold. First, we show that our RL-based method achieves an impressive level of generalization. We remind that we trained `RFL` on random instances of just two anchors and evaluated the method on unseen instances with up to 600 anchors. The scalability of our method to handle hundreds of agents relies on two key aspects: the design of our environment and the use of Multi-Agent Deep Reinforcement Learning (MADRL). As the number of agents increases, more potential conflicts can occur. However, our local-global reward structure motivates the agents to minimize these conflicts collectively. Notably, our design does not involve any explicit communication channel

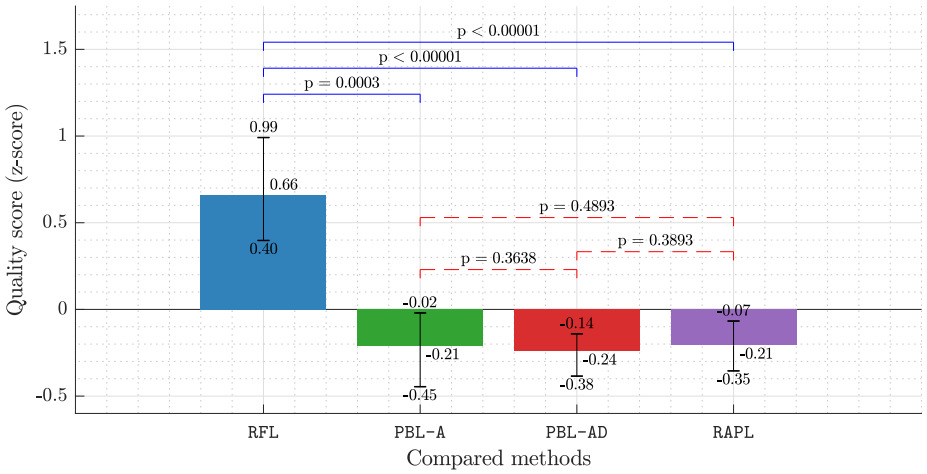


Figure 4.5: Results of the user study. The chart shows quality scores accompanied by 95% confidence intervals. Statistically significant differences between the pairs of methods are denoted by solid blue brackets, with the corresponding p-values reported above them. Conversely, the red dashed brackets represent the pairs of methods without evidence of statistically significant differences.

among the agents. Instead, a shared policy implicitly encourages collaborative behavior, resulting in predictably coordinated actions among agents. Even so, `RFL` outperformed the compared methods in the category of completeness, and at the same time, the user study participants preferred `RFL` over all compared methods. By this fact, we demonstrate the power of machine learning techniques and their capability to surpass the hand-crafted algorithms.

Second, we show that greedy methods frequently produce suboptimal solutions concerning completeness. The quantitative results shown in Section 4.4.1 and the comparison of the examined methods in Table 4.4 provide evidence supporting this claim. For example, in instance 45, `RAPL` and `PBL-A` left four anchors unlabeled. Even `PBL-AD`, with the benefit of distant labels, did not find complete labeling and left two anchors unlabeled. In contrast, `RFL` produced complete adjacent labelings for all these instances.

Third, we observe a trade-off between optimality and computation time demands. All the previous methods we examined, `PBL-A`, `PBL-AD`, and `RAPL`, can be computed faster but at the expense of an incomplete solution. In contrast, the proposed `RFL` method is several orders of magnitude slower but, on the other hand, provides results with a much higher level of completeness. Therefore, the examined methods, `PBL-A`, `PBL-AD`, and `RAPL`, are suitable for interactive applications where incompleteness is not critical (interactive visualizations with the ability to zoom). On the other hand, `RFL` is better suited for cases where the labeling can be computed in advance, and completeness is essential (*e.g.*, cartographic maps, technical drawings, medical atlases). We argue that our approach can serve better than the other examined methods to aid professional illustrators. We believe that future research based on `RFL` can further mitigate the gap between optimality and speed.

Finally, even though we have primarily focused on the elimination of label overlap in this work, the RL framework is much more versatile. It enables the integration of other metrics into the reward function, allowing one to tailor the solution to specific tasks and opening up numerous opportunities for further improvements and research. The flexibility

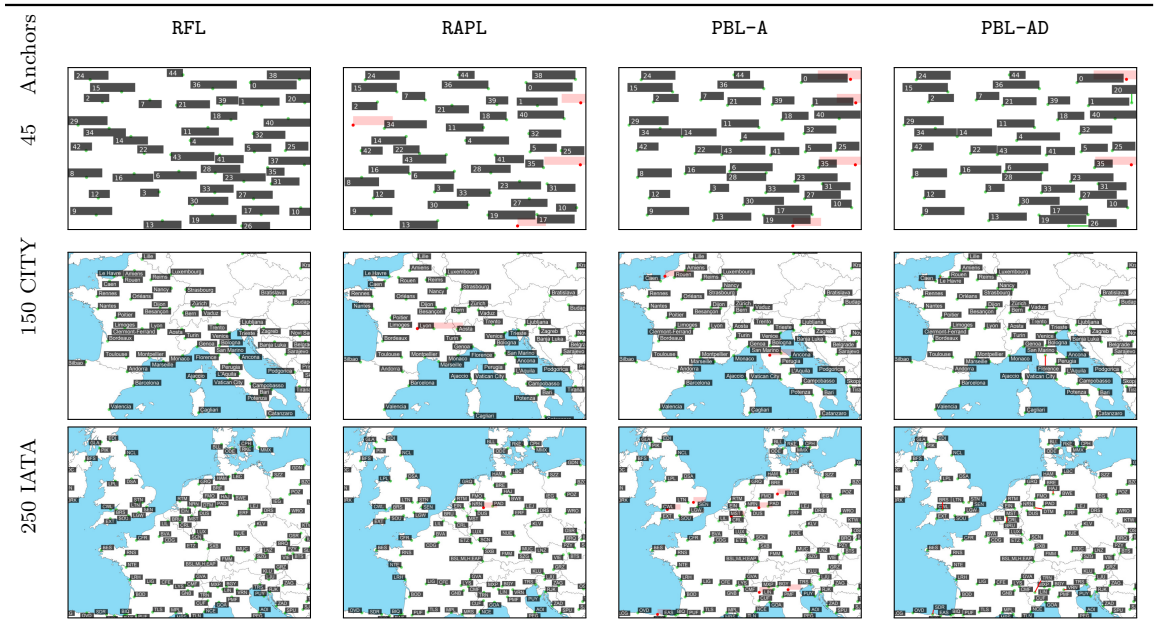


Table 4.4: Visual comparison of the examined methods for selected instances from the proposed dataset. In addition, we provide real-world instances of IATA airport codes with 250 anchors and CITY names with 150 anchors based on data obtained from Open Street Maps. For the latter two, we cropped the result to focus on differences among methods. Original results can be found in Appendix B. The green dot represents the anchor (*i.e.*, point feature). The gray rectangle symbolizes the body of the label itself. The red dot describes an anchor that was not labeled by the given method. The red rectangle illustrates the dimensions of the missing label. We stress that we intentionally added all the missing labels to the visualization for illustrative purposes only, and their origins are not the outcome of the method itself.

solidifies the potential of RL in solving complex spatial decision-making tasks like label placement, promising exciting advancements in the visualization field.

4.5 Limitations

We are aware of several flaws and limitations of the proposed method. First and foremost, `RFL` is currently limited to finding only binary solutions. As a result, the method can find either complete conflict-free labeling or a complete but conflict-present solution (*i.e.*, label-label or label-anchor at the screen bound). However, the flexible design of the `AdjacentPFLEnv` environment allows one to define new actions or entirely redefine the existing ones. In future work, adding a further indicator action, which decide whether to place the label or not, could address the binary limitation of the current method. Similarly, the reward objectives could be extended to minimize the number of unlabeled anchors. To this end, we intend to make the `AdjacentPFLEnv` publicly available to further support the research in this domain.

Computation time is another area for improvement of the proposed method. We showed in Section 4.4.1 that `RFL` is magnitudes of order slower than the compared methods. Further examination revealed that the computation and collection of observations contribute significantly to the computation time, mainly due to expensive ray casting and imperfect code

optimization. However, we believe that in future work, the limitation can be addressed, for instance, by using a graph representation of possible conflicts known as a conflict graph in the observation. Nevertheless, we leave the question open for future research.

4.6 Summary

In this chapter, we introduced the first Multi-Agent Deep Reinforcement Learning formulation of the adjacent-point-feature labeling problem. To facilitate the label placement policy training, we developed `AdjacentPFLEnv`, an environment where agents collect experiences – sense the state of the environment via proposed observation modalities, perform actions, and receive feedback in the form of proposed reward. Furthermore, we designed an efficient yet straightforward feedforward neural network architecture with less than half of a million parameters to model the agent's policy and estimate the value function. We show that our approach significantly outperforms previous hand-crafted methods designed by human experts in the number of placed labels and perceived quality. Additionally, we would like to encourage the labeling community towards standardized evaluation, a long-used machine learning practice. To this end, we are proposing a new benchmark dataset to facilitate the comparison of label placement methods, as most of the method codes remain unpublished or proprietary.

Chapter 5

Perceptual Prioritization of Point-Feature Label Positions

Point-feature label placement primarily deals with the maximization problem – aiming to position labels for the maximum number of point features, also called *anchors*, possible within a given set. This task is often constrained by the *fixed-position* model, which restricts label placements to a limited number of predefined positions around a point feature. This limitation necessitates using a systematic order of preference for these positions, which we term as Position Priority Order (PPO). The PPO ranks potential label positions according to predetermined priorities, guiding the selection process.

However, a closer review of existing literature reveals a disconcerting lack of consensus in PPOs as described in detail in Section 2.4.2. Various authors have ascribed different priorities to the same label positions, often without a clear or unified rationale. Priorities have historically been based on typographic and cartographic conventions or printer capabilities, with varying degrees of justification and consistency across the literature, leading to a fragmented understanding of optimal label placement practices. The identified inconsistency highlights a gap in the field and underlines the necessity for an empirically grounded methodology that reflects user perceptions and preferences. Nevertheless, recent PFLP approaches [93, 102, 138] continue to rely on traditional PPOs. Interestingly, commercial products such as Google Maps, TomTom, and Mapbox tend to use non-traditional PPOs for reasons that have not been reported. This discrepancy between academic research and commercial practice further emphasizes the urgent need for updated, user-validated PPOs.

Our research introduces Perceptual Position Priority Order (`PerceptPPO`), a user-centered methodology that seeks to redefine the prioritization of point-feature label positions based on users' perceptual and cognitive preferences rather than traditional conventions. Through this effort, we aim to establish a new standard in automatic label placement that prioritizes user experience, paving the way for more intuitive and accessible map designs and setting a precedent for future research in the domain. Our main contributions are summarized as follows:

(**1**) We propose a comprehensive review of existing literature on Position Priority Orders (PPOs), analyzing existing PPOs and highlighting the missing consensus based on user preferences.

(**2**) We introduce Perceptual Position Priority Order (`PerceptPPO`), a novel, user-centered prioritization of point-feature label positions that prioritizes user perceptions and

preferences over traditional conventions supported by a global user study involving nearly 800 participants from 48 countries.

(**3**) We uncover the optimal label density for maps, an aspect seldom explored in prior research. Our research shows that users prefer an overall label coverage of 12.5% on blind maps.

(**4**) We provide analysis demonstrating the superior user preference of `PerceptPPO` over existing PPOs, reinforcing its potential for improving map design, user experience in cartographic applications, and other types of visualizations.

5.1 Perceptual Position Priority Order (`PerceptPPO`)

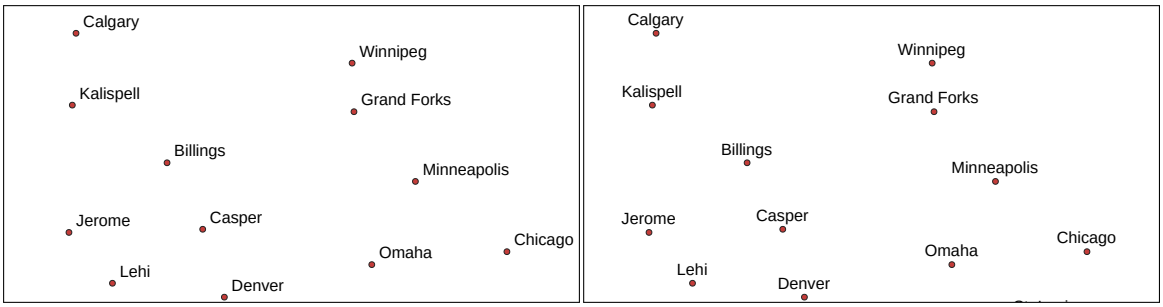
Due to non-existing consensus on the ranking of label positions within cartography and GIS, we aim to create PPO rooted in user perception. Typographic and cartographic conventions used in previous works are valuable, but originate from several decades-old practices that may not align with modern user needs. Herein, we detail the empirical user study that underpins the Perceptual Position Priority Order (`PerceptPPO`) and lay the groundwork for a comparative analysis that underscores its efficacy.

5.1.1 Data

We randomly selected 30 locations worldwide. We excluded any locations on the sea or ocean and those with latitudes greater than -60 degrees to exclude Antarctica due to its sparse population. Each location served as the center of an area, defined by the location and a zoom level ranging from 5 (approximately the size of Europe) to 10 (roughly the size of Luxembourg), rendered as a vector SVG image at a size of 1305×1025 pixels. Settlements with more than 500 habitants, obtained from GeoNames, specifically Cities 500[1], within these areas were used as anchors and were sorted by population size. Then we filtered only anchors such that all the 8-positions around are available in any configurations of labels without any conflict (we examined the occurrence of conflict for all anchors over bounding boxes containing all eight positions of labels). If an area contained fewer than 20 anchors, we discarded it in favor of another area. Finally, we acquired 30 areas indexed from 0 to 29 at zoom levels 5 to 8, with 20 to 54 anchors.

Subsequently, we rendered each area eight times, placing all labels in one of the eight corresponding positions relative to the anchor: top-right (TR), top (T), top-left (TL), left (L), bottom-left (BL), bottom (B), bottom-right (BR), and right (R). This process yielded $30 \times 8 = 240$ blind maps featured with a white background, red anchors, and corresponding labels. See example in Figure 5.1 and Appendix C for additional renders. In order to understand user preferences for label positions in relation to the anchor, we opted for using blind maps to eliminate all other factors potentially influencing the judgment of the label placement, such as patterns and vivid colors in the map background. Each cartographic rule for label placement [33, 82, 146, 158] focuses on a specific factor, such as avoiding overlaps, ensuring proper label alignment, or selecting appropriate font sizes and colors. Therefore, we believe that these factors can be analyzed separately.

[1] http://download.geonames.org/export/dump/



(a) TR (b) T

Figure 5.1: Example of map area rendered consistently using TR and T position across all point features.

5.1.2 Procedure

We employed the Two-Alternative Forced Choice (2AFC) approach to determine the position priority order based on the actual perception of users, which we call Perceptual Position Priority Order (`PerceptPPO`). Considering the eight label positions under examination, we have $\binom{8}{2} = 28$ pairwise comparisons on an area. Consequently, the entire evaluation consists of $30 \times 28 = 840$ pairwise comparisons to cover all of them once. In order to alleviate potential fatigue among the participants during the study, we allocated only three areas to each participant, resulting in a batch of $3 \times 28 = 84$ pairwise comparisons. Therefore, 10 participants were required to cover all pairwise comparisons once.

We engaged Mechanical Turk workers and university students to conduct the evaluation. Participation in the evaluation was voluntary, and the compensation for Mechanical Turk workers was set to match the average compensation rate of other requesters on Amazon Mechanical Turk. Initially, the participants were introduced to the experiment, informed about its duration, and that the provided data would be collected and used for research purposes. Subsequently, we asked about their country of residence, age, gender, and education. Each participant was allowed to take only one batch of 84 pairwise comparisons to mitigate the carry-over effect. During the evaluation, the pairs of area and label position were distributed randomly but uniformly between the left and right sides of the shown comparison pair. Participants were shown two maps sequentially, each depicting the same area but with different positions of labels relative to the anchors. Notably, the position of a label was kept consistent for all points within a single map; see Figure 5.1. Participants were asked to select the map they prefer from each presented pair. At the end of the evaluation, the participants were allowed to leave an additional note about anything regarding the experiment.

5.1.3 Statistical Analysis

To derive insights from the pairwise comparison data of n objects, we transform the data into the $n \times n$ preference matrix P for each participant individually. Each element p_{ij} in the matrix indicates the number of times that method i was selected over method j by a participant. The conversion allows for a preference analysis, facilitating the identification of PPO patterns across participants in the study.

Coefficient of Consistency ζ

Assessing whether the particular participant can form a reliable judgment of the quality under examination is crucial when using paired comparison. Kendall and Babington [89] proposed *coeficient of consistency* ζ to measure the consistency based on the transitivity of participant's choices. For example, when evaluating three objects: A, B, and C, a participant might choose that $A > B$ (A is preferred to B), $B > C$ and $C > A$. In this case, the *triad* is called *circular* and the pair comparison inconsistent. The $\zeta = 1$ if there are no circular triads / no inconsistencies. On the other hand, when the number of circular triads/inconsistencies increases, the ζ decreases towards zero. Inconsistency can arise due to incompetence, participant's attention changes during the evaluation, or the examined objects being too alike. The definition of ζ can be simplified as $\zeta = 1 - \frac{T}{T_{max}}$, where T is the observed number of circular triads, T_{max} is the maximum number of circular triads. For more details; see Kendall and Babington [89] and David [45]. We employ the implementation proposed by Wickelmaier and Schmid [177], which also provides the expected number of circular triads $E(T)$ when choices are made at random.

Coefficient of Agreement u

While the coefficient of consistency provides a measure of consistency within participants, the *coefficient of agreement* u introduced by Kendall and Babington [89] measures the variety of choices among m participants. Complete agreement $u = 1$ is achieved when all participants make identical choices for all pairs. In other words, the half p_{ij} of the overall preference matrix $\mathbf{P} = \sum_{k \in (0,m)} P_k$ is equal to the number of participants m, while the other half is a zero. On the other hand, the minimum agreement occurs when the preference for each pair is equally divided among participants. Specifically, this happens when $p_{ij} = \frac{m}{2}$ if m is even or when $p_{ij} = \frac{m \pm 1}{2}$ otherwise. Correspondingly, minimum coefficient of agreement is $u_{\min} = \frac{-1}{m-1}$ or $u_{\min} = \frac{-1}{m}$. For cases in between the u range, Kendall and Babington [89] defines u as

$$u = \frac{2 \sum_{i \neq j} \binom{p_{ij}}{2}}{\binom{m}{2}\binom{n}{2}} - 1. \tag{5.1}$$

The statistical significance of u with the null hypothesis that all participants choose the preferences randomly (or there is no agreement among participants) can be approximated by χ^2 variate as described in David [45]. Again, we employ the implementation proposed by Wickelmaier and Schmid [177].

Pairwise Comparison Model

To transform the preference matrices to quality scores, we employ Thurstone's statistical judgment model proposed by Thurstone [165], as recommended by Tsukida and Gupta [166] and Pérez-Ortiz and Mantiuk [139]. The model assumes that the quality score of object A is a Normal random variable $A \sim \mathcal{N}(q_A, \sigma_A^2)$ where mean q_A is assumed to be the true quality score, and σ_A^2 is the variance. Similarly, for object $B \sim \mathcal{N}(q_B, \sigma_B^2)$. Normal distribution captures the fact that different participants have various preferences regarding the quality of examined objects (inter-participant variance). Moreover, participants' preferences are also likely to change when they repeat the same evaluation (intra-participant variance). We apply Thurstone's Case V model, which assumes that a Normal distribution can explain inter- and intra-participant variance. At the same time, the variance σ^2 describes the

uncertainty and is the same for all examined objects (in our example $\sigma_A^2 = \sigma_B^2$) while the correlation ρ among objects is zero ($\rho_{AB} = 0$). The difference between the two Normal distributions is again Normal distribution $A-B \sim \mathcal{N}(q_{AB}, \sigma_{AB}^2)$. Without loss of generality, we can assume that variance $\sigma_A^2 = \sigma_B^2 = \frac{1}{2}$ so that $\sigma_{AB}^2 = \sigma_A^2 + \sigma_B^2 - 2\rho_{AB}\sigma_A\sigma_B = 1$ which corresponds to standard Normal distribution. The quality difference estimation of two objects A and B $\hat{q}_{AB}$ is then defined as

$$\hat{q}_{AB} = \Phi^{-1}\left(\frac{\mathbf{P}_{A,B}}{\mathbf{P}_{A,B} + \mathbf{P}_{B,A}}\right), \tag{5.2}$$

where $\mathbf{P}_{A,B}$ is the value from the preference matrix $\mathbf{P}$ for $A > B$, and $\Phi^{-1}(x)$ is the inverse CDF of standard Normal distribution that can be interpreted as *z-score* as it represents the distance of x from the mean in units of the standard deviation.

To determine quality scores of m objects, Tsukida and Gupta [166] recommend using the maximum likelihood estimate (MLE). To this end, we employ the MLE implementation of Pérez-Ortiz and Mantiuk [139], which also includes confidence interval estimation based on random sampling with replacement and methodology to perform a two-tailed test of the null hypothesis "There is no difference among examined objects." at a significance level of $\alpha = 0.05$.

5.1.4 Online Study Precautions

When dealing with online study, there is always a risk of ingenue responses and result fabrication. Therefore, to address these pitfalls, we conducted a pilot study with 50 participants who are qualified as Master Mechanical Turk Workers and consistently demonstrated high accuracy in performing various Human Intelligence Tasks (HITs). The pilot results serve as a calibration group to determine an evaluation's statistics, evaluation duration, time spent on the introduction page, time spent filling out the survey, response time for individual pairs, participant's coefficient of consistency, and balance of choosing the left or right option. Afterward, we made the study available to a broader range of Mechanical Turk Workers while following the general recommendations: HIT approval rate 95%, number of approved HITs > 2000, and restricted repetition of study by one worker. Additionally, we implemented Google reCAPTCHA to reduce the risk of bot fabrication, mitigating bot activity and ensuring that participants are genuine. Moreover, to assess and control the data quality, we used the statistics from the calibration group to eradicate workers of insufficient quality that significantly deviated from the standard deviation. We intentionally did not automate the elimination process to interpolate the measured statistics (deviations from the calibration group) while considering the expected number of circular triads $E(T)$ and workers' feedback, allowing a more nuanced understanding of worker performance and potential issues within the tasks. In particular, we identify participants as potentially inconsistent if $T < E(T)$ is observed in at least two of the three areas assigned for their evaluation. Once the HIT was approved, it became part of the calibration group, and all statistics for this group were recalculated.

To gain an even deeper understanding, we employed Smartlook[2] to analyze workers' behavior while working on the evaluation by manually reviewing activity recordings. By doing so, we identified three pitfalls: (**1**) some workers were likely modifying JavaScript to alter the behavior of the study, (**2**) workers often copied the text of the task/questions pre-

[2] www.smartlook.com

sumably because they did not understand the text and were translating it, (**3**) some workers disregarded the instructions and tried to complete the task as quickly as possible.

To address these issues, we (**1**) implemented obfuscation to prevent manipulation of JavaScript, (**2**) integrated Google Translate into the study and slightly modified the instructions to accommodate non-native English speakers better, and (**3**) introduced a mechanism where a participant must first review the pair and spend a minimum of 5 seconds selecting their preferred option.

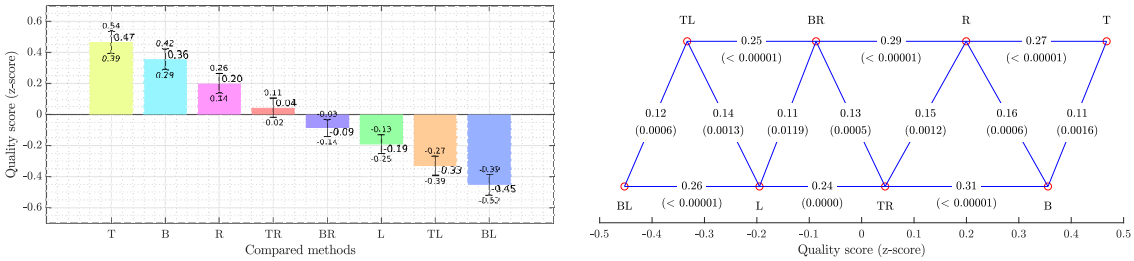
5.1.5 Results

We eliminated participants who exhibited inconsistencies in their responses as described in Section 5.1.4. Following this data refinement, we were left with a total of 225 participants with a dropout of 27%. These participants generated 18,900 pairwise comparisons. Therefore, on average, each comparison pair was evaluated by approximately 23 different participants.

We apply methodology as described in Section 5.1.3 to compute the quality z-score and assess statistical significance at the significance level of $\alpha = 0.05$ to evaluate the null hypothesis H_0^1: "There is no clear user preference among the label positions." Our initial findings of aggregated preferences suggested that label positions could be ordered by the perceptual preferences of participants as follows: T > B > R > TR > BR ≥ L > TL > BL. A statistically significant difference was found between all pairs of label positions except for the BR ≥ L pair. Our finding also harmonizes with PPO proposed by Scheuerman *et al.* [151], who claims that the L > TL > BL order is preferred among participants. In order to determine statistical significance for the BR ≥ L pair, we engaged an additional 104 participants who were specifically asked to respond to the (BR, L) pair of positions. Each participant was presented with a single pair for each area, resulting in 30 pairwise comparisons per participant. This approach led to a total of 3,120 new pairwise comparisons specific to the pair of (BR, L) positions. After the addition of the new comparisons, the final results, as depicted in Figure 5.2(a), show a statistically significant difference between all pairs of label positions, as illustrated by Figure 5.2(b). Therefore, we can reject the null hypothesis H_0^1 and claim that there is a clear preference of label positions in the order given by z-scores.

In total, we engaged 329 participants for this study, comprising of 217 males and 112 females. A majority of the participants hailed from the USA (155), followed by Czech Republic (88), India (35), and Slovakia (29). The most common age range among the participants was 20-30 years, with 159 individuals falling into this category. Regarding educational qualifications, the highest number of participants held bachelor's degrees (119), followed by high school diplomas (117), and master's degrees (65). On average, participants completed a batch of 84 pairwise comparisons in 5 minutes and 56 seconds, with a standard deviation of 3 minutes and 4 seconds. For the additional comparisons specifically acquired to determine statistical significance for the (BR, L) pair, the average completion time was 2 minutes and 4 seconds, with a standard deviation of 1 minute and 10 seconds.

The overall average consistency ζ across participants is 0.67 ($SD = 0.29$, $MD = 0.75$), which indicates that they were fairly consistent in their decisions and the consistency is reasonably leveled for each map area. For more details see Table C.9 in Appendix C. The overall coefficient of agreement $u = 0.12$ (min $u = -0.001$) reveals relatively low agreement among participants, although with the p-value $= 0$ clearly indicates that we can reject the



(a) (b)

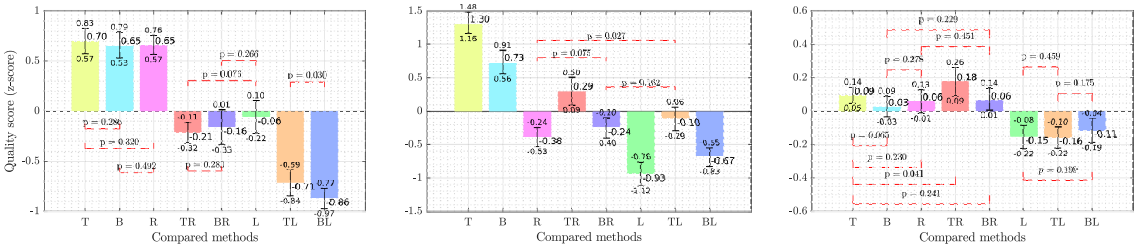
Figure 5.2: Results of the `PerceptPPO` user study. Chart (a) depicts quality z-scores and 95% confidence intervals. Chart (b) shows a triangle plot visualizing significant differences between label position preferences as proposed by Pérez-Ortiz and Mantiuk [139]. Each red circle represents a label position, and the lines indicate significant differences between pairs. Solid blue lines represent statistically significant differences. The edge values show the absolute difference in z-scores between the compared label positions, with the p-values denoted in brackets. The label positions are plotted along the x-axis, with alternating y-axis offsets for clarity.

null hypothesis H_0^2: "There is no agreement among participants" at $\alpha = 0.05$ and conclude there is indeed statistically significant agreement among participants.

However, the relatively low overall coefficient of agreement $u = 0.12$ suggests that there might be underlying patterns or segments within the participant data that are not immediately apparent from the aggregated overall results. Therefore, we use hierarchical clustering applying Ward's minimum variance method to uncover these patterns and provide a more nuanced interpretation of the data [71, 83]. We identified three participant clusters as shown in Figure 5.3. Even though the p-value for the coefficient of agreement u within clusters is sometimes greater than $\alpha = 0.05$ for individual areas, which disallows us to reject the null hypothesis H_0^2 for several areas, especially in Cluster 3, aggregation of the choices over all areas leads to p-values lower than $\alpha = 0.05$ for all clusters. Therefore, among all clusters, there is indeed statistically significant agreement among participants. Cluster 1 ($N = 93$) shown in Figure 5.3(a) with mean consistency $\zeta_1 = 0.833$ ($SD = 0.179$, $MD = 0.900$) and fairly high agreement $u_1 = 0.335$ (min $u_1 = -0.004$), comprises participants that show strong preference in central positions T, B, R, and partly L over to corner positions BL, TL, TR, and BR. Cluster 2 ($N = 41$) depicted in Figure 5.3(b) with mean consistency $\zeta_2 = 0.844$ ($SD = 0.155$, $MD = 0.900$) also shows considerable agreement $u_2 = 0.370$ (min $u_3 = -0.008$) and contains participants that strongly favor label positions T, B, TR as opposed to L, BL, R, BR, and TL. Cluster 3 ($N = 91$) presented in Figure 5.3(c) with mean consistency $\zeta_3 = 0.435$ ($SD = 0.259, MD = 0.350$) and relatively low agreement $u = 0.021$ (min $u = -0.004$) includes participants that are uncertain in their preferences but lean towards TR position.

5.2 Evaluation of `PerceptPPO`

In order to evaluate the established `PerceptPPO` and compare it with the other PPOs typesetted in bold in Table 2.4, we have conducted a series of the following evaluations. We do not include the position modifiers SL (Slightly Left) and SR (Slightly Right) in our study. Initially, the modifiers were introduced due to limitations in grid printing, specifically for



(a) **C1** ($N = 93$, $\zeta_1 = 0.833$, $u_1 = 0.335$, $p = 0$) (b) **C2** ($N = 41$, $\zeta_2 = 0.844$, $u_2 = 0.37$, $p = 1.5e^{-258}$) (c) **C3** ($N = 91$, $\zeta_3 = 0.435$, $u_3 = 0.021$, $p = 2.7e^{-25}$)

Figure 5.3: Identified clusters of users in `PerceptPPO` study with depicted quality z-scores and 95% confidence intervals. The red dashed brackets, with the corresponding p-values reported above, represent the PPOs pairs without evidence of statistically significant differences at $\alpha = 0.05$ using the two-tail test. Conversely, between the PPOs pairs without brackets are detected statistically significant differences.

labels with an even number of letters that could not be precisely centered. However, with current advancements in typesetting, the necessity for these auxiliary positions has become obsolete. Furthermore, we examine at most eight positions to reduce the complexity of a time-demanding pairwise comparison study, as the additional positions are relatively uncommon. The additional criteria for the selection were the number of citations (applied for Brewer [33], Christensen and Marks [39], Slocum [158], Imhof [82]), the founder aspect (Imhof [82], Yoeli [185]), and similarity with `PerceptPPO` (Zoraster [193]). In the following text, we typeset these using `typewriter` font to denote the corresponding PPO and abbreviate the PPO proposed by Christensen and Marks only by the first author's name (`Christensen`).

5.2.1 Evaluation 1: Label Density

We need map renders to evaluate and compare the `PerceptPPO` with existing PPOs. Again, we intend to create a blind map to eliminate factors potentially influencing the judgment of the label placement other than its position relative to the anchor. However, by doing so, we faced a question. How many labels should be presented in such a map area? We reviewed existing cartographic books and found that, surprisingly, just a handful of works studied this topic [80]. Therefore, we conducted a dedicated experiment to see users' preferences on label density.

Data

We selected ten populated areas (0, 4, 5, 6, 9, 12, 13, 17, 27, 29), providing a wide range of possible label density samples from areas employed in the `PerceptPPO` study. We limited our selection to ten areas because not all 30 locations described in Section 5.1.1 had sufficient populations to sample the varying levels of label density required for the experiment. Additionally, reducing the number of areas streamlined the experiment, making it more manageable and efficient for participants. For each area, we produced maps with varying density of labels which the participants can compare. We have several requirements on each map: (**1**) The Global Label Density (GLD) of the whole map M is lower than a given threshold LD_{thr} and (**2**) to prevent local dense clusters of labels in highly populated areas,

the Local Label Density (LLD) of local neighborhood of each anchor has to be lower than the given threshold LD_{thr}. Both these densities are expressed in percentages.

We define LLD for each anchor α in map M, such that the anchor is the center of a tile T of size 256×256, which is the typical size of web-based raster maps. The LLD is defined as:

$$LLD(\alpha) = \frac{\sum_{a \in M} L(a) \cap T}{A(T)}, \tag{5.3}$$

where a is an anchor, T is the square tile, $L(a)$ is the rectangle enclosing the label text corresponding to an anchor a, and $A(T)$ is the area of the tile. If the anchor is in the map M positioned such that T is not entirely in the map M, we perform a minimal shift of the tile T to be entirely in the map M. Similarly, we define GLD for each map M as

$$GLD(M) = \frac{\sum_{a \in M} L(a) \cap M}{A(M)}. \tag{5.4}$$

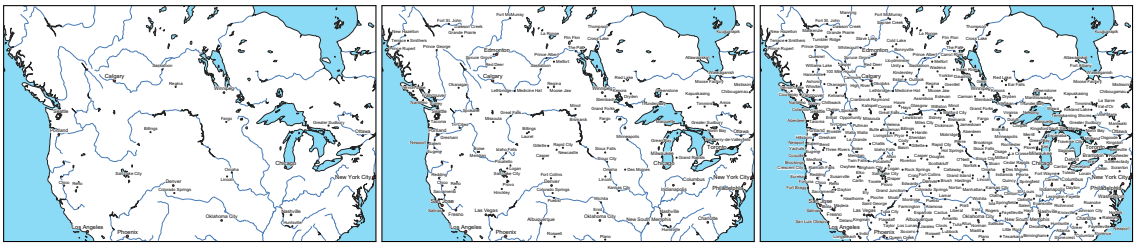
We order the cities in the given map area by population size, from most populated to least populated, and process them iteratively. We sequentially try to add anchors and labels to the map, starting with the largest population and proceeding to the smallest. The anchor α is added to map M at position p from PPO if all of the following conditions are met: (**1**) Anchor α is in the map M, (**2**) the label L of anchor α is entirely in the map M, (**3**) the label L of anchor α is not overlapping any anchor, (**4**) the label L of anchor α is not overlapping any already placed label, (**5**) the local label density LLD of anchor α is lower than LD_{thr}, (**6**) the global label density GLD of the map M is lower than LD_{thr}. After no other anchors can be labeled according to the above rules, we recompute the final LLD ($LLDF$) for each placed anchor as we do not update $LLD(a_1)$, when label $L(a_2)$ protrudes from T_2 into T_1 and label $L(a_2)$ is placed after $L(a_1)$. Therefore, $LLDF$ can be slightly above LD_{thr}.

We aggregate $LLDF$ within each map M by calculating the median and mean values across all anchors $\alpha \in M$, capturing overall statistics and allowing for consistent comparison of local label density across different maps. Specifically, these aggregated values are calculated as follows:

$$\widehat{LLDF}(M) = \text{median}_{\alpha \in M}(LLDF(\alpha)) \tag{5.5}$$

$$\overline{LLDF}(M) = \text{mean}_{\alpha \in M}(LLDF(\alpha)) \tag{5.6}$$

To prepare various samples of selected areas with various levels of label densities, we repeatedly rendered the area with $LD_{\text{thr}} = (2.5\% - 40\%, \text{step size } 2.5\%) \cup (45\%, 50\%, 75\%, 100\%)$. The upper part is more sparse because, with increasing levels of label density, the renders became perceptually similar. For example, only a few labels create differences between 40% and 42.5% or 50% and 55%. Therefore, we increased the step size to 5% from 40% and to 25% from 50%. If any rendered maps of a given area were the same, we kept only unique map renders. Following this procedure, we rendered ten selected areas (see Figure 5.4 and for more examples, refer to Appendix C) for selected PPOs from Table 2.4. Again, We used the same data source for cities as in Section 5.1.1. We employed the city labels in three different sizes, as suggested by various guidelines [33, 98, 146, 158] , because more than three categories are perceived with difficulty according to Robinson *et al.* [146]. As suggested by Tyner [167], we split the population into three intervals: ($< 500,000$); (500,001–1,000,000); and ($> 1,000,000$); and apply three different font sizes of 11pt, 13pt, and 15pt, differing by 2pt as recommended by Robinson *et al.* [146].

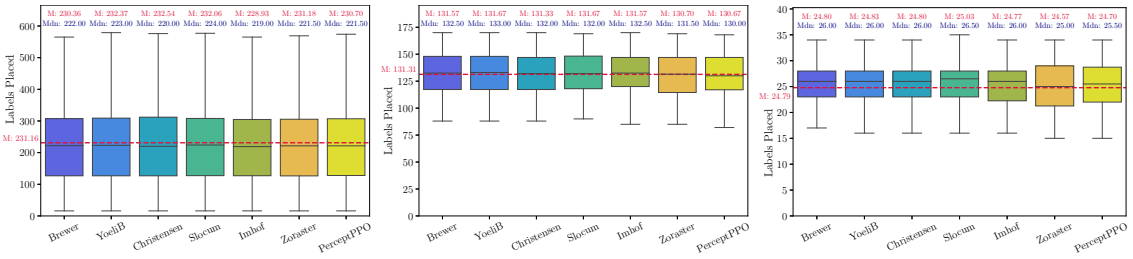


(a) $LD_{\text{thr}} = 2.5\%$, $\widehat{LLDF} = 3.4\%$, $\overline{LLDF} = 3.9\%$, $GLD = 2.5\%$

(b) LD_{thr}=12.5%, $\widehat{LLDF} = 16\%$, $\overline{LLDF} = 15.8\%$, $GLD = 12.5\%$

(c) $LD_{\text{thr}} = 100\%$, $\widehat{LLDF} = 50.1\%$, $\overline{LLDF} = 46.7\%$, $GLD = 31.3\%$

Figure 5.4: Renders of map area 0 using `PerceptPPO` at various values of LD_{thr} employed in the evaluation of label density described in Section 5.2.1.



(a) All LD_{thr} aggregated at 1305 × 1025

(b) $LD_{\text{thr}} = 12.5\%$ at 1305 × 1025

(c) $LD_{\text{thr}} = 12.5\%$ at 652 × 512

Figure 5.5: (a) The box plot of placed labels for each examined PPO aggregated for all values of LD_{thr} at 1305 × 1025. (b) The box plot of placed labels for label density $LD_{\text{thr}} = 12.5\%$ at 1305 × 1025. (b) The box plot of placed labels for label density $LD_{\text{thr}} = 12.5\%$ at 652 × 512.

For the evaluation of label density, we are not using a blind map, but we included the borders of continents and water bodies in terms of rivers, lakes, seas, and oceans (see Figure 5.4 and Appendix C). By doing so, we aimed to provide the participants with information about where the cities may occur and where they might not (*e.g.*, a lake or ocean) while minimizing the factors affecting the participants' decision process. We used the data provided by Natural Earth[3], specifically land, lakes and reservoirs, rivers, and lake central lines. We selected data from available scales at 1:50,000,000 due to the trade between precision and image size as we employ SVG vector format at a size of 1305 × 1025 pixels.

Data Verification

We measured the number of positioned labels for all PPOs to ensure that anchors' spatial configurations within selected areas do not discriminate against or favor a particular PPO. The Figure 5.5(a) and Figure 5.5(b) show that the number of placed labels is very similar for all PPOs. Additionally, we explored the adherence of label placements to their designated priorities and examined how increasing label densities influence the label placement. For each selected PPO outlined in Table 2.4, we have measured the probability of labels occupying specific positions and observed how these probabilities shift with escalating label densities.

[3] https://www.naturalearthdata.com/downloads/

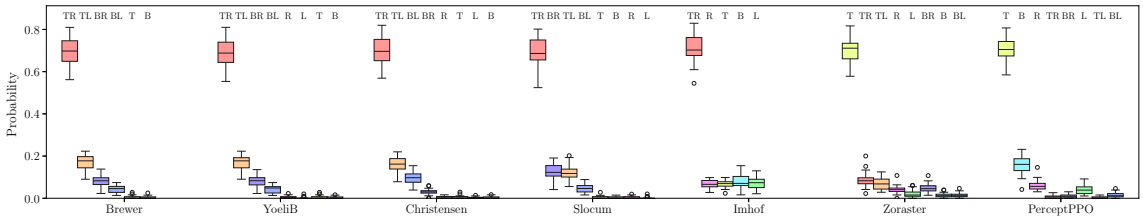


Figure 5.6: The probability of a label being placed at a given position over all 30 locations rendered as described in Section 5.2.1 with label density $LD_{\text{thr}} = 12.5\%$.

As expected, labels are most likely to be positioned in their highest-priority position. The probability of the label being placed at the second highest priority was significantly lower, decreasing progressively for lower-priority positions (refer to Figure 5.6 and Figure C.15 in Appendix C for additional insights into label density impacts). Nonetheless, some patterns deviated from this trend. We observed a striking deviation with priorities of Imhof [82], where labels at lower densities were equally likely to occupy the second to fifth priority positions. However, as label density increased, this likelihood inverted, favoring lower-priority positions, realigning the priority sequence from the intended TR, R, T, B, L to TR, L, B, T, R for maps with greater label density. This density-dependent realignment in positioning probabilities is detailed in Figure C.15 in Appendix C.

We also noted exceptions in the PPOs set by Slocum [158], Zoraster [193], and our `PerceptPPO`. For the priorities of `Slocum`, labels at higher densities are more likely to be placed at TL than at BR, contrary to the intended order BR, TL. Similarly, for the priorities of `Zoraster`, the likelihood of position BR is higher than for L or R, contrary to intended order R, L, BR. Likewise, `PerceptPPO` demonstrated a higher placement likelihood at L over TR or BR. Appendix C further illustrates these individual probability patterns across different PPOs and their relation to label density in Figure C.15.

Procedure

Participants could access the study directly via a web application or indirectly via the Mechanical Turk interface, which embedded the same web app within its environment. Participation in the evaluation was voluntary, and the compensation for Mechanical Turk workers was set to match the average compensation rate of other requesters on Amazon Mechanical Turk. The web application included an introduction, a survey, an evaluation, and a feedback section. The introduction defined the participant's task and how to use the application. Within the evaluation we presented rendered maps of selected areas in randomized order. Each participant was assigned a single PPO for all ten areas. Each participant was allowed to participate only once to mitigate the carry-over effect. For each area, we preloaded all renders with various label density levels. Using the slider, the participants were asked to "choose a label density they found comfortable without being overwhelmed by the amount of information." The leftmost position was the lowest label density, and the rightmost position was the highest density. The participant must spend at least 5 seconds carefully selecting the preferred label density and explore the full range of label densities. When they complete all ten areas, we provide them the option to provide any feedback.

Results

We eliminated participants who exhibited inconsistencies within their responses across different areas and ones that deviated more than two standard deviations from the mean of $\overline{LLDF}$ and evaluation duration, as described in Section 5.1.4. After this data refinement, we engaged 110 participants from Mechanical Turk with a relatively high dropout of 45%, which we attribute to a spike of inattentive workers according to the review of activities recorded by Smartlook. Eliminated workers consistently set the slider in the right or leftmost position, which is highly unlikely to be the preferred position according to the control group.

We found the following overall statistics on the preference of label density threshold LD_{thr} from participants: median(LD_{thr}) = 12.5% (μ = 14.85, SD = 7.22%). We determined the preferred LD_{thr} using the median, as it provides a robust measure that minimizes the impact of potential outliers. The preferred LD_{thr} = 12.5% corresponds to the median(GLD) = 12.5% (μ = 14.54%, SD = 6.59%), and a median of 150 labeled anchors (μ = 159, SD = 82). To accurately reflect the trends in participants' opinions on the local label density, we calculated the overall preference statistics from the aggregated median $\widehat{\overline{LLDF}}$ and mean $\overline{LLDF}$, respectively. Specifically, the preferred LD_{thr} = 12.5% corresponds to median($\widehat{\overline{LLDF}}$) = 16.3% ($\mu$ = 17%, SD = 7.35%).

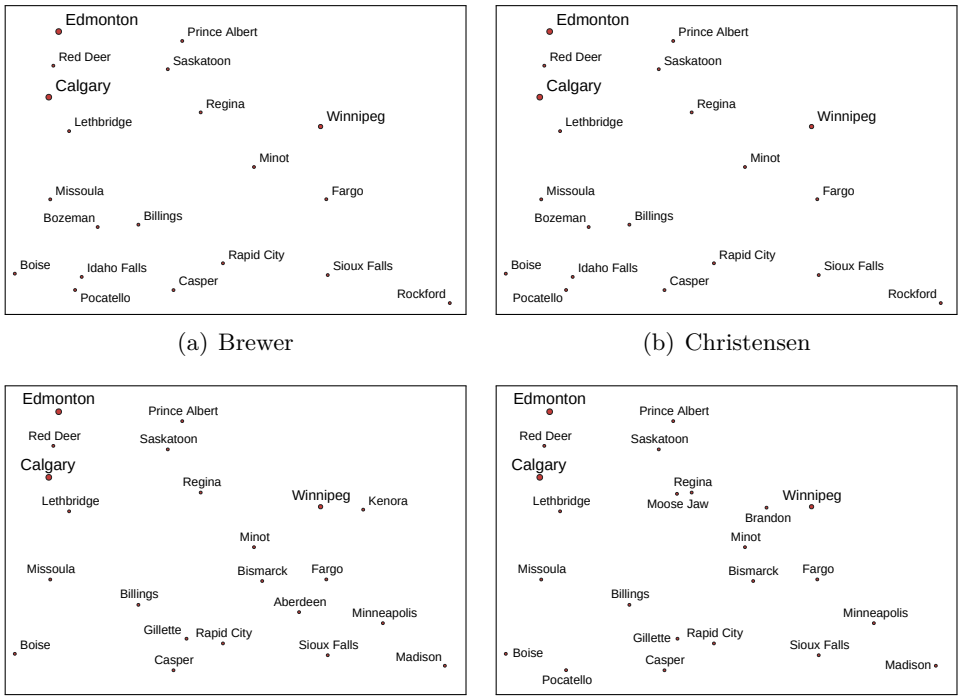
Upon conducting an Analysis of Variance (ANOVA) to investigate the effect of different PPOs on the LD_{thr}, our results indicated no significant differences across the various PPOs. Specifically, the ANOVA test, utilizing a Type II sum of squares approach, yielded an F-statistic of 1.660 with a corresponding p-value of 0.127. The p-value suggests that we found no statistically significant differences in LD_{thr} values among PPOs at significance level $\alpha = 0.05$. The finding implies that the examined PPOs do not significantly affect the LD_{thr} value, reinforcing the idea that the variations observed in LD_{thr} across different PPOs might be attributed to random chance rather than inherent differences in PPOs. Therefore, our analysis supports the conclusion that the selection of PPOs does not significantly influence the preferred LD_{thr}.

5.2.2 Evaluation 2: Comparison of PPOs

To compare the proposed `PerceptPPO` with existing PPOs, we conducted an evaluation that follows findings from the evaluation of label density experiment described in Section 5.2.1. We aim to validate that the `PerceptPPO` is preferred when the label cannot always be placed in a single position, unlike the `PerceptPPO` experiment described in Section 5.1.

Data

In this evaluation, we employ the same approach as described in Section 5.2.1 with LD_{thr} = 12.5%. However, this time, we again use a blind map to eliminate factors potentially influencing the judgment of the label placement other than its position relative to the anchor. We have also reduced the size of renders to half 652 × 512, which means that the average number of shown anchors and labels is 24.79 (SD = 5.91), contrary to 131.31 (SD = 21.01) for 1305 × 1025; see Figure 5.5 and Appendix C for more details. We found out that comparing map pairs with 131.31 labels on average is overly demanding and cannot be completed in a reasonable time. Therefore, we chose to reduce the size of the map area while still maintaining the LD_{thr} = 12.5% but with 24.79 presented labels on average; see Figure 5.7. In this evaluation, we also use all 30 locations worldwide as described in



(a) Brewer (b) Christensen

(c) Zoraster (d) PerceptPPO

Figure 5.7: Renders of map area 0 using various PPOs at $LD_{\text{thr}} = 12.5\%$ employed in comparison of PPOs described in Section 5.2.2.

Section 5.1.1. Other data characteristics hold with Section 5.2.1, except the background is blank as in Section 5.1.1.

Data Verification

We measured the number of placed labels for all PPOs again to ensure that anchors' spatial configurations within selected areas do not discriminate against or favor a particular PPO. The Figure 5.5(c) show that the number of placed labels is very similar for all PPOs. Finally, we explored the adherence of label placements to their designated priorities for the data used in the evaluation (652×512 size of renders at $LD_{\text{thr}} = 12.5\%$). Refer to Figure C.16 in Appendix C for more details. We found the same deviations for the priority orders of `Imhof`, `Zoraster`, and `PerceptPPO` as for the data with 1305×1025 size of renders at $LD_{\text{thr}} = 12.5\%$ (see Section 5.2.1).

Procedure

We applied the 2AFC paradigm to determine the preferred PPO based on the perception of users. Considering the seven PPOs under examination, we have $\binom{7}{2} = 21$ pairwise comparisons on an area. Consequently, the entire evaluation consists of $30 \times 21 = 630$ pairwise comparisons to cover all of them once. In order to alleviate potential fatigue among the participants during the study, we allocated only three areas to each participant, resulting in a batch of $3 \times 21 = 63$ pairwise comparisons. Therefore, 10 participants were required to cover all pairwise comparisons once. We engaged Mechanical Turk workers and university students to perform the evaluation. The rest of the procedure is the same as

for Section 5.1.2 except, participants were shown two maps sequentially, each depicting the same area but with different PPOs.

Results

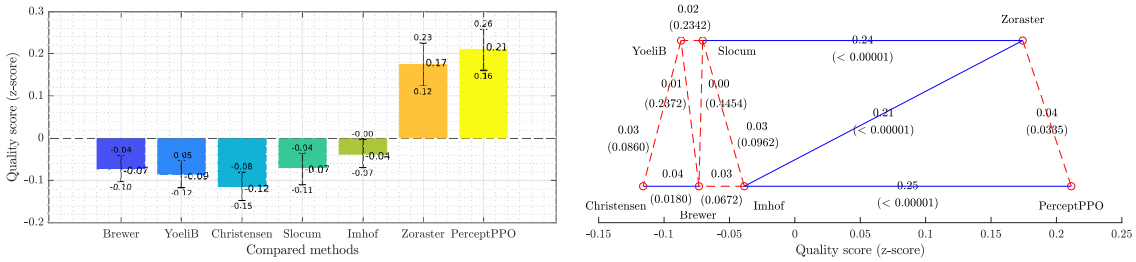
We eliminated participants with inconsistent responses as described in Section 5.1.4. Following this data refinement, we were left with a total of 352 participants (179 females and 173 males) with a dropout of 24%.

A majority of the participants hailed from the USA (221), followed by India (54), Czech Republic (32), Brazil (7), and United Kingdom (5). The most common age range among the participants was 20-30 years, with 98 individuals falling into this category, followed by 31-40 with 93 participants, 41-50 with 77 participants, 51-60 with 47 participants, > 60 with 35 participants and 2 participants bellow 20. Regarding educational qualifications, the highest number of participants held bachelor's degrees (163), master's degrees (76), followed by high school diplomas (67), community college education (38), doctoral's degrees (7), and elementary education (1). On average, participants completed a batch of 63 pairwise comparisons in 10 minutes and 46 seconds ($SD = 3$ minutes and 13 seconds). The participants generated 22,236 pairwise comparisons. Therefore, on average, each comparison pair was evaluated by approximately 35 different participants.

We apply methodology as described in Section 5.1.3 to compute the quality z-score and assess statistical significance at the significance level of $\alpha = 0.05$ to evaluate the null hypothesis H_0^3: "There is no clear user preference among the examined PPOs." The overall results, as depicted in Figure 5.8, show a statistically significant difference between PPO groups G_{T} ={`PerceptPPO`, `Zoraster`} and G_{TR} ={`Brewer`, `YoeliB`, `Christensen`, `Slocum`, `Imhof`}. Specifically, the statistical analysis indicates that there is no significant difference within the groups G_{T}, and G_{TR}, but there is a significant difference between these two groups. Figure 5.8(b) demonstrates that `PerceptPPO` and `Zoraster` consistently outperform other PPOs, justifying their grouping. Therefore, we can reject the null hypothesis H_0^3 and claim that there is a clear preference of PPOs only between groups G_{T} and G_{TR}. In other words, participants perceived PPOs in G_{T} significantly better than in G_{TR}. The outcome validates the findings of our `PerceptPPO` study described in Section 5.1 that participants prefer label position T at the first place of PPO as proposed by `Zoraster` and `PerceptPPO`. Interestingly, the following label positions do not seem essential for perceiving the quality of PPOs.

The overall average consistency ζ across participants is 0.59 ($SD = 0.23$, $MD = 0.57$), which indicates that they were fairly consistent in their choices. The consistency remained reasonably uniform across each map area; details are provided in Table C.10 in Appendix C. The overall coefficient of agreement $u = 0.013$ (min $u = -0.0009$) with the p-value $= 7.362e^{-55}$ clearly shows that we can reject the null hypothesis H_0^4: "There is no agreement among participants" at $\alpha = 0.05$ and conclude there is indeed statistically significant agreement among participants.

However, the relatively low overall coefficient of agreement $u = 0.013$ suggests that there might be underlying patterns or segments within the participant data that are not immediately apparent from the aggregated overall results. Therefore, we apply the same hierarchical clustering technique as in Section 5.1.5 to uncover these patterns and provide a more nuanced interpretation of the data. We identified three participant clusters as shown in Figure 5.9. Even though the p-value for the coefficient of agreement u within clusters is sometimes greater than $\alpha = 0.05$ for individual areas, which disallows us to reject the



(a) (b)

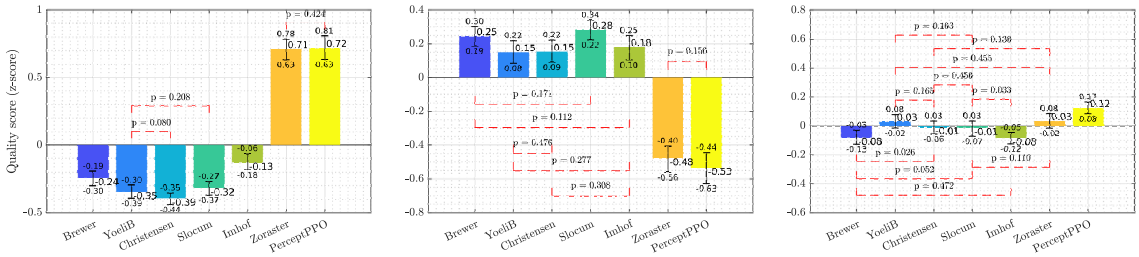
Figure 5.8: Overall comparison of PPOs. Chart (a) depicts quality score and 95% confidence intervals. Chart (b) shows a triangle plot visualizing significant differences between PPO preferences as proposed by Pérez-Ortiz and Mantiuk [139]. Each red circle represents a PPO, and the lines indicate significant differences between pairs. Solid blue lines represent statistically significant differences, while the dashed red lines indicate non-significant ones. The edge values show the absolute difference in z-scores between the compared PPOs, with the p-values denoted in brackets. The PPOs are plotted along the x-axis, with alternating y-axis offsets for clarity.

null hypothesis H_0^4 for several areas, aggregation of the choices over all areas leads to p-values lower than $\alpha = 0.05$ for all clusters. Therefore, among all clusters, there is indeed statistically significant agreement among participants. Detailed results supporting found clustering, including statistics and cluster compositions, can be found in Table C.10 in Appendix C.

Cluster 1 ($N = 134$) depicted in Figure 5.9(a) with mean consistency $\zeta_1 = 0.659$ ($SD = 0.232$, $MD = 0.714$) shows fairly high agreement $u_1 = 0.218$ (min $u_1 = -0.002$, p-value$= 0$) and contains participants that prefer PPOs in group $G_{\mathrm{T}} = \{\texttt{PerceptPPO}, \texttt{Zoraster}\}$. Cluster 2 ($N = 82$) shown in Figure 5.9(b) with mean consistency $\zeta_2 = 0.612$ ($SD = 0.223$, $MD = 0.643$) and slightly lower agreement $u_2 = 0.131$ (min $u_2 = -0.004$, p-value$= 1.25e^{-134}$), comprises participants that prefer PPOs in group $G_{\mathrm{TR}} = \{\texttt{Brewer}, \texttt{YoeliB}, \texttt{Christensen}, \texttt{Slocum}, \texttt{Imhof}\}$ contrary to PPOs in group G_{T}. Cluster 3 ($N = 136$) presented in Figure 5.9(c) with mean consistency $\zeta_3 = 0.502$ ($SD = 0.210$, $MD = 0.500$) and relatively low agreement $u_3 = 0.005$ (min $u_3 = -0.002$, p-value$= 7.63e^{-6}$) includes participants that are uncertain in their preferences but slightly incline towards PPOs in group G_{T} over PPOs in group G_{TR}.

5.3 Discussion

The `PerceptPPO` study showcases the potential of a user-centered approach to enhancing the principles of label placement in cartography and GIS. The high coeficient of consistency ζ observed among the participants, with an overall mean of 0.67 and a median of 0.75, underscores the reliability of user judgments in determining the perceptual preference order of label positions around point features. Notably, the study established a clear overall preference order: top (T) > bottom (B) > right (R) > top-right (TR) > bottom-right (BR) > left (L) > top-left (TL) > bottom-left (BL). Moreover, the analysis of the participants' feedback revealed that their choices were driven primarily by the functionality of the label placement rather than aesthetics. Many participants highlighted that their preferences were based on the ease of associating labels with their anchors. Specifically, labels positioned



(a) **C1** ($N = 134$, $\zeta_1 = 0.659$, $u_1 = 0.218$, $p = 0$) (b) **C2** ($N = 82$, $\zeta_2 = 0.612$, $u_2 = 0.131$, $p = 1.2e^{-134}$) (c) **C3** ($N = 136$, $\zeta_3 = 0.502$, $u_3 = 0.005, p = 7.6e^{-6}$)

Figure 5.9: Identified clusters of users in a comparison study of PPOs with depicted quality z-scores and 95% confidence intervals. The red dashed brackets, with the corresponding p-values reported above, represent the PPOs pairs without evidence of statistically significant differences at $\alpha = 0.05$ using the two-tail test. Conversely, between the PPOs pairs without brackets are detected statistically significant differences.

directly on top or bottom of the anchor were most effective in facilitating the association, with the top position being particularly favored. This finding challenges traditional conventions and suggests a shift towards prioritizing labels above or bellow of point features for improved user experience.

The evaluation of label density offers a brief view into how users perceive and prefer the density of labels on maps – the median of the final preferred local label density at 16.3% reflects the preferred label density in the local region of a map. The median of global label density of 12.5% suggests that users favor a moderate overall label presence, enough to inform without being too crowded. Analysis of participants' feedback further supports our findings, as they consistently highlighted preference selection driven by functionality over aesthetics. Participants reported that when label density was too low, they struggled to orient themselves in the area due to a lack of contextual information. Conversely, when label density was too high, they found the maps cluttered, making it challenging to associate labels with the corresponding cities. Therefore, achieving the right balance is crucial for creating user-friendly maps that facilitate easy navigation and understanding.

Finally, the comparison between `PerceptPPO` and existing PPOs reveals that `PerceptPPO`'s perceived quality surpasses the traditional PPOs and slightly outperforms `Zoraster`. Moreover, our comparison of `PerceptPPO` with `Zoraster` suggests that the initial position within a PPO plays a crucial role, while subsequent positions have less impact on overall user perception.

5.4 Limitations

While the study presents a statistically significant ordering of label positions, it also uncovers distinct clusters of user preferences, revealing the complexity of perceptual prioritization. We used generic blind maps that, on the one hand, minimize the degree of freedom and enable precise evaluation of preferences, but on the other hand, it is essential to acknowledge that other elements of a map could also influence user perception. Additionally, this study did not account for semantic considerations – such as the placement of city labels across state borders or the positioning of coastal city labels towards water bodies, as highlighted by Imhof [82]. Preferences might also vary based on the map's intended use and audience, from

military to recreational purposes. The demographic aspects may similarly affect the preferences for languages with right-to-left or top-to-bottom scripts. Additionally, participants' familiarity with the presented area may influence their preferences. Notably, the study did not delve into the functional aspect of label placement, including ease of information search and overall readability. Despite stringent participant screening and measures to ensure data integrity, the inherent variables associated with the uncontrolled online environment could introduce unseen biases.

The outcomes of our study lay a solid foundation, yet further research is needed to explore the aforementioned factors. Future work should investigate how semantic rules, map purpose, target audience preferences, and functional aspects such as readability interplay with the perceptual prioritization of label positions to develop more nuanced guidelines for map design.

5.5 Summary

In this chapter, we introduced Perceptual Position Priority Order (`PerceptPPO`), fundamentally reviewing the point-feature label placement by prioritizing user preferences over traditional conventions. Engaging nearly 800 participants globally, we have established a user-preferred ordering of label positions along the feature point that challenges and diverges from the conventional top-right towards the top position. Moreover, we performed an additional study to find users' preferred label density, a domain narrowly studied in prior literature. According to the results, users prefer 12.5% of the generic map to be overall covered by labels. Finally, the comparative analysis underscores `PerceptPPO`'s superiority in perceived quality against traditional PPOs, particularly highlighting the significance of the first label position's role in user perception. Our empirical study marks a significant step toward more intuitive and user-centered map designs, emphasizing the importance of aligning label placement visualization practices with actual user expectations.

Chapter 6

Conclusions

The primary objective of this dissertation was to advance the field of automatic label placement by developing innovative techniques to address the inherent challenges of both internal and external label placement within complex visualizations. This research was driven by the need to enhance the efficiency and user perception of label placements across various applications. Specifically, the research concentrated on three key areas: achieving temporally stable and visually coherent boundary label placement, exploring the relevance of machine learning in internal label placement to enhance the completeness of labeled data, and optimizing label positioning by incorporating perceptual insights.

Research Outcomes

Chapter 2 provided a comprehensive review of existing techniques in automatic label placement, highlighting the limitations of current methods in handling dynamic environments and maintaining visual coherence. The review also identified that while significant progress has been made, challenges remain in effectively balancing computational efficiency with the quality of label placement. Additionally, despite the widespread adoption of Position Priority Order (PPO), a systematic ranking of potential label positions around a point feature, there remains no consensus within the labeling community on the optimal approach to determining the PPO.

The research presented in Chapter 3 introduced novel approaches for the temporally stable optimization of boundary labeling in panoramic visualizations. The proposed optimization techniques successfully minimized label movement across frames in dynamic visualizations, thereby enhancing visual coherence and reducing cognitive load on users. The experimental results demonstrated the effectiveness of these approaches in maintaining label stability without compromising readability or clarity.

Chapter 4 explored the relevance of machine learning for internal point-feature label placement. The proposed methods significantly improved the completeness of labeled data, particularly in scenarios involving complex and dense data. By leveraging reinforcement learning, the proposed technique could dynamically adapt to varying data distributions and optimize label positions, outperforming traditional heuristic-based methods.

The final Chapter 5 focused on perceptual prioritization in label positioning relative to the point feature. This research established a user-preferred ordering of label positions, challenging the conventional preference for the top-right position by demonstrating a stronger user preference for labels placed directly above the feature point. Additionally, an

exploratory study was conducted to identify users' preferred label density, which has been scarcely explored in existing literature. The findings confirmed that integrating perceptual insights into the label placement process significantly enhances overall user experience, leading to more intuitive and effective visualizations.

Limitations

While this research has made substantial contributions to the field of automatic label placement, several limitations must be acknowledged. The proposed methods, particularly for boundary label optimization in panoramic visualizations, are more complex and challenging to implement compared to existing techniques. While allowing for greater flexibility, this complexity may pose challenges in practical applications, especially in scenarios requiring real-time processing. The deep reinforcement learning approach, although effective in improving label completeness, is computationally intensive. The time required for computation, particularly in scenarios with a large number of labels, is a limitation that could hinder its application in real-time systems. The approach is also limited to binary solutions, either achieving complete conflict-free labeling or allowing conflicts. The perceptual prioritization study provided valuable insights into user preferences for label positions, but it also revealed the complexity and variability of these preferences. The study's use of generic blind maps minimized variables, but future research should consider how additional map elements, semantic rules, and the map's intended audience and purpose might influence user preferences. Moreover, demographic factors and the functional aspects of label placement, such as ease of information search and readability, were not fully explored.

Future Work

Future research in the field of automatic label placement can explore several promising directions that build on the work presented in this dissertation. One intriguing avenue is the development of mixed-label placement models that effectively operate in three-dimensional spaces. This challenge involves integrating the concepts of internal point-feature label placement, typically used in top-down views, with external label placement, which is more suitable for side views. Such a model would be particularly useful in applications where terrains or structures need to be visualized from multiple perspectives, such as in geographic information systems, interactive hiking guides, or urban planning. Combining these two approaches presents a unique challenge, as it requires developing algorithms that can dynamically adjust label placements based on the viewer's perspective, ensuring that labels remain legible and correctly associated with their corresponding features. This problem, which has not been widely addressed in the research community, offers significant potential for innovation.

The deep reinforcement learning approach explored in this dissertation has proven effective in specific contexts but remains limited to binary solutions, either achieving complete conflict-free label placement or allowing conflicts. Future research could focus on optimizing the computational aspects of this approach. For instance, employing more efficient data structures, such as conflict graphs, could streamline identifying and resolving label conflicts. Improving the optimization process in the proposed way could enhance the practicality of Reinforced Labels (RFL), particularly in real-time applications where computational efficiency is critical.

Another critical area for future research is enlarging our understanding of user perception in label placement. While the perceptual prioritization study conducted in this dissertation provided valuable insights, it also highlighted the complexity and variability of user preferences. Future studies could explore how different factors, such as the content of a map, the intended use of the visualization, and the demographic characteristics of the user base, influence label placement preferences. Moreover, developing a user-centered metric for evaluating the perceived quality of label placements could be a valuable tool for improving label placement algorithms. Such a metric could be derived from crowd-sourced data used to train neural networks that predict the perceived quality of a label placement, thereby providing user-centered feedback during the label placement process. This approach could close the loop in the label placement process, using these predictions as metrics to guide the placement of labels in a way that aligns more closely with user expectations and preferences.

Finally, it is essential to acknowledge that the quality of label placement is not entirely and precisely defined. While cartographic guidelines provide a framework for minimizing conflicts and ensuring clear associations between labels and features, many of these rules are based on best practices rather than empirical user studies. As such, the objectives used in label placement optimization may not always be applicable across different domains. Therefore, future research should focus on developing flexible label placement techniques that can easily incorporate various criteria and adapt to the specific needs of diverse applications. Flexibility is crucial for ensuring that label placement techniques remain relevant and effective across a wide range of use cases. The proposed methods in this dissertation have been designed with this flexibility in mind, driven by mathematical definitions that can be easily altered or extended. However, further collaboration between visual designers, cartographers, and computer scientists is needed to refine these techniques and enhance their applicability in real-world scenarios.

This research journey has been marked by significant challenges and meaningful discoveries. In conclusion, while this dissertation has made substantial strides in advancing the field of automatic label placement, much work remains to be done. By addressing the limitations identified and exploring the new directions outlined above, further research can contribute to the development of more robust, efficient, and user-centered visualizations that better serve the needs of diverse users and applications.

Bibliography

[1] Agarwal, P. K.; van Kreveld, M. and Suri, S. Label placement by maximum independent set in rectangles. *Computational Geometry*, 1998, vol. 11, no. 3, p. 209–218. ISSN 0925-7721.

[2] Agresti, A. and Coull, B. A. Approximate is better than exact for interval estimation of binomial proportions. *The American Statistician*. Taylor & Francis, 1998, vol. 52, no. 2, p. 119–126. ISSN 0003-1305.

[3] Ahn, J. and Freeman, H. A program for automatic name placement. *Cartographica*, 1984, vol. 21, p. 101–109. ISSN 0317-7173.

[4] Ali, K.; Hartmann, K. and Strothotte, T. Label layout for interactive 3d illustrations. *Journal of the WSCG*, 2005, vol. 13, no. 1, p. 1–8. ISSN 1213-6972.

[5] Altman, D. G. and Bland, J. M. How to obtain the P value from a confidence interval. *BMJ*. BMJ Publishing Group Ltd, 2011, vol. 343. ISSN 0959-8138.

[6] Alvim, A. C. and Taillard, É. D. POPMUSIC for the point feature label placement problem. *European Journal of Operational Research*, 2009, vol. 192, no. 2, p. 396–413. ISSN 0377-2217.

[7] Apple Inc. *Map of Central Europe generated by MapKit JS*. 2024. Available at: https://developer.apple.com/maps/mapkitjs/. Accessed: 2024-06-20.

[8] Arulkumaran, K.; Deisenroth, M. P.; Brundage, M. and Bharath, A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 2017, vol. 34, p. 26–38. ISSN 1053-5888.

[9] Azuma, R. and Furmanski, C. Evaluating label placement for augmented reality view management. In: *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality*. 2003, p. 66–75. ISBN 978-0-7695-2006-3.

[10] Balata, J.; Čmolík, L. and Mikovec, Z. On the selection of 2D objects using external labeling. In: *CHI '14*. 2014. ISBN 978-1-4503-2473-1.

[11] Barth, L.; Gemsa, A.; Niedermann, B. and Nöllenburg, M. On the readability of boundary labeling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9411, p. 515–527. ISSN 1611-3349.

[12] Barth, L.; Niedermann, B.; Nöllenburg, M. and Strash, D. Temporal map labeling: a new unified framework with experiments. In: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery, 2016. SIGSPACIAL '16. ISBN 978-1-4503-4589-7.

[13] Been, K.; Daiches, E. and Yap, C. Dynamic map labeling. *IEEE Transactions on Visualization and Computer Graphics*, 2006, vol. 12, no. 5, p. 773–780. ISSN 1077-2626.

[14] Been, K.; Nöllenburg, M.; Poon, S. H. and Wolff, A. Optimizing active

ranges for consistent dynamic map labeling. *Computational Geometry: Theory and Applications*, 2010, vol. 43, no. 3, p. 312–328. ISSN 0925-7721.

[15] Bekos, M. A.; Cornelsen, S.; Fink, M.; Hong, S. H.; Kaufmann, M. et al. Many-to-one boundary labeling with backbones. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2013, 8242 LNCS, p. 244–255. ISSN 0302-9743.

[16] Bekos, M. A.; Kaufmann, M.; Potika, K. and Symvonis, A. Multi-stack boundary labeling problems. *WSEAS Transactions on Computers*, 2006, vol. 5, no. 11, p. 2602–2607. ISSN 1109-2750.

[17] Bekos, M. A.; Kaufmann, M.; Symvonis, A. and Wolff, A. Boundary Labeling: Models and Efficient Algorithms for Rectangular Maps. In: *Graph Drawing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 49–59. ISBN 978-3-540-31843-9.

[18] Bekos, M. A.; Kaufmann, M.; Symvonis, A. and Wolff, A. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry: Theory and Applications*, 2007, vol. 36, no. 3, p. 215–236. ISSN 0925-7721.

[19] Bekos, M. A.; Kaufmann, M.; Nöllenburg, M. and Symvonis, A. Boundary Labeling with Octilinear Leaders. In: *SWAT '08: Proceedings of the 11th Scandinavian workshop on Algorithm Theory*. Springer, 2008, p. 234–245. ISBN 978-3-540-69900-2.

[20] Bekos, M. A.; Niedermann, B. and Nöllenburg, M. External labeling techniques: A taxonomy and survey. *Computer Graphics Forum*, 2019, vol. 38, no. 3, p. 833–860. ISSN 1467-8659.

[21] Bekos, M. A.; Niedermann, B. and Nöllenburg, M. *External Labeling: Fundamental Concepts and Algorithmic Techniques*. 1st ed. Springer Cham, 2021. 1 p. ISBN 978-3-031-01481-9.

[22] Benichou, M.; Gauthier, J. M.; Girodet, P.; Hentges, G.; Ribiere, G. et al. Experiments in mixed-integer linear programming. *Mathematical Programming*, 1971, vol. 1, no. 1, p. 76–94. ISSN 0025-5610.

[23] Benkert, M.; Haverkort, H.; Kroll, M. and Nöllenburg, M. Algorithms for multi-criteria one-sided boundary labeling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2008, 4875 LNCS, no. 3, p. 243–254. ISSN 0302-9743.

[24] Berg, M. D. and Gerrits, D. H. Approximation algorithms for free-label maximization. *Computational Geometry: Theory and Applications*, may 2012, vol. 45, p. 153–168. ISSN 0925-7721.

[25] Bertini, E.; Rigamonti, M. and Lalanne, D. Extended excentric labeling. *Computer Graphics Forum*, 2009, vol. 28, no. 3, p. 927–934. ISSN 1467-8659.

[26] Bhore, S.; Li, G. and Nöllenburg, M. An Algorithmic Study of Fully Dynamic Independent Sets for Map Labeling. In: *28th Annual European Symposium on Algorithms (ESA 2020)*. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, vol. 173, p. 19:1–19:24. Leibniz International Proceedings in Informatics (LIPIcs). ISBN 978-3-95977-162-7.

[27] Bixby, E. R.; Fenelon, M.; Gu, Z.; Rothberg, E. and Wunderling, R. MIP: Theory and Practice — Closing the Gap. In: *System Modelling and Optimization*. Boston, MA: Springer US, 2000, p. 19–49. ISBN 978-0-387-35514-6.

[28] Bliek, C.; Bonami, P. and Lodi, A. Solving Mixed-Integer Quadratic Programming problems with IBM-CPLEX : a progress report. *Proceedings of the Twenty-Sixth RAMP Symposium*, 2014, M, p. 171–180.

[29] Bobák, P.; Čmolík, L. and Čadík, M. Video Sequence Boundary Labeling with Temporal Coherence. In: *Proceedings of Computer Graphics International 2019*. Cham: Springer International Publishing, 2019, p. 40–52. CGI 2019. ISBN 978-3-030-22514-8.

[30] Bobák, P.; Čmolík, L. and Čadík, M. Temporally stable boundary labeling for interactive and non-interactive dynamic scenes. *Computers & Graphics*, 2020, vol. 91, p. 265–278. ISSN 0097-8493.

[31] Bose, P.; Carmi, P.; Keil, J.; Mehrabi, S. and Mondal, D. Boundary labeling for rectangular diagrams. In: *16th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2018*. Germany: Schloss Dagstuhl- Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, June 2018, p. 12:1–12:14. Leibniz International Proceedings in Informatics, LIPIcs. ISBN 978-3-95977-068-2.

[32] Bose, P.; Mehrabi, S. and Mondal, D. Faster Multi-sided One-Bend Boundary Labelling. In: *WALCOM: Algorithms and Computation*. Springer Science and Business Media Deutschland GmbH, 2021, 12635 LNCS, p. 116–128. ISBN 978-3-03068210-1.

[33] Brewer, C. *Designing better maps: a guide for GIS users*. 2nd ed. ESRI Press, 2015. ISBN 978-1-58948-440-5.

[34] Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J. et al. *OpenAI Gym*. arXiv, 2016.

[35] Bylinskii, Z.; Kim, N. W.; O'Donovan, P.; Alsheikh, S.; Madan, S. et al. Learning visual importance for graphic designs and data visualizations. In: *UIST 2017 - Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 2017, p. 57–69. ISBN 978-1-4503-4981-9.

[36] Canese, L.; Cardarilli, G. C.; Nunzio, L. D.; Fazzolari, R.; Giardino, D. et al. Multi-agent reinforcement learning: A review of challenges and applications. *Applied Sciences (Switzerland)*, 2021, vol. 11. ISSN 2076-3417.

[37] Chen, D. S.; Batson, R. G. and Dang, Y. *Applied Integer Programming: Modeling and Solution*. 1st ed. John Wiley & Sons, 2011. ISBN 978-0-470-37306-4.

[38] Chen, Z.; Wang, Y.; Wang, Q.; Wang, Y. and Qu, H. Towards automated infographic design: Deep learning-based auto-extraction of extensible timeline. *IEEE Transactions on Visualization and Computer Graphics*. IEEE Computer Society, january 2020, vol. 26, p. 917–926. ISSN 1941-0506.

[39] Christensen, J. and Marks, J. An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Transactions on Graphics (TOG)*, 1995, vol. 14, no. 3, p. 203–232. ISSN 1557-7368.

[40] Čmolík, L. and Bittner, J. Layout-Aware Optimization for Interactive Labeling of 3D Models. *Computers & Graphics*, 2010, vol. 34, no. 4, p. 378–387. ISSN 0097-8493.

[41] Čmolík, L.; Pavlovec, V.; Wu, H.-Y. and Nöllenburg, M. Mixed Labeling: Integrating Internal and External Labels. *IEEE Transactions on Visualization and Computer Graphics*, 2022, vol. 28, no. 4, p. 1848–1861. ISSN 1941-0506.

[42] Cook, A. C. and Jones, C. B. A Prolog Rule-Based System for cartographic Name Placement. *Computer Graphics Forum*, 1990, vol. 9, no. 2, p. 109–126. ISSN 0167-7055.

[43] Cravo, G. L.; Ribeiro, G. M. and Lorena, L. A. N. A greedy randomized adaptive search procedure for the point-feature cartographic label placement. *Computers and Geosciences*, 2008, vol. 34, no. 4, p. 373–386. ISSN 0098-3004.

[44] Da Col, S.; Ciucanu, R.; Soare, M.; Bouarour, N. and Amer Yahia, S. DashBot: An ML-Guided Dashboard Generation System. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2021, p. 4696–4700. CIKM '21. ISBN 978-1-4503-8446-9.

[45] David, H. *The method of paired comparisons*. 2nd ed. C. Griffin, 1988. Griffin's statistical monographs & courses. ISBN 978-0-85264-290-0.

[46] Deng, Y.; Bao, F.; Kong, Y.; Ren, Z. and Dai, Q. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*. IEEE, 2017, vol. 28, no. 3, p. 653–664. ISSN 2162-2388.

[47] Dent, D. B.; Torguson, J. S. and Hodler, T. W. *Cartography: Thematic Map Design*. 6th ed. McGraw-Hill Education, 2009. 336 p. ISBN 978-0-07-294382-5.

[48] Doddi, S.; Marathe, M. V.; Mirzaian, A.; Moret, B. M. E. and Zhu, B. Map labeling and its generalizations. In: *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*. 1997, p. 148–157. ISBN 978-0-89871-390-9.

[49] Doerschler, J. S. and Freeman, H. A rule-based system for dense-map name placement. *Commun. ACM*. New York, NY, USA: Association for Computing Machinery, jan 1992, vol. 35, no. 1, p. 68–79. ISSN 0001-0782.

[50] Ebinger, L. R. and Goulette, A. M. Noninteractive Automated Names Placement for the 1990 Decennial Census. *Cartography and Geographic Information Systems*. Taylor & Francis, 1990, vol. 17, no. 1, p. 69–78. ISSN 1523-0406.

[51] Edmondson, S.; Christensen, J.; Marks, J. and Shieber, S. A General Cartographic Labelling Algorithm. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 1996, vol. 33, p. 13–24. ISSN 0317-7173.

[52] Eurosport. *Olympics Games Paris: Women's Double Sculls Rowing Semi-Final*. 2024. Available at: https://www.youtube.com/watch?v=WuH1jVppQNE. Accessed: 2024-08-13.

[53] Fekete, J. D. and Plaisant, C. Excentric labeling: Dynamic neighborhood labeling for data visualizaition. In: *CHI '99*. 1999, p. 512–519. ISBN 978-0-201-48559-2.

[54] Formann, M. and Wagner, F. A packing problem with applications to lettering of maps. In: *Proceedings of the Seventh Annual Symposium on Computational Geometry*. New York, NY, USA: Association for Computing Machinery, 1991, p. 281–288. SCG '91. ISBN 978-0-89791-426-0.

[55] Freeman, H. An expert system for the automatic placement of names on a geographic map. *Information Sciences*, 1988, vol. 45, no. 3, p. 367–378. ISSN 0020-0255.

[56] Garey, M. R. and Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. 1st ed. New York, NY, USA: W. H. Freeman & Co., 1979. ISBN 978-0-7167-1045-5.

[57] Gedicke, S.; Bonerath, A.; Niedermann, B. and Haunert, J. Zoomless Maps: External Labeling Methods for the Interactive Exploration of Dense Point Sets at a Fixed Map Scale. *IEEE Transactions on Visualization and Computer Graphics*. Los Alamitos, CA, USA: IEEE Computer Society, feb 2021, vol. 27, no. 02, p. 1247–1256. ISSN 1941-0506.

[58] Gedicke, S.; Arzoumanidis, L. and Haunert, J. H. Automating the external

placement of symbols for point features in situation maps for emergency response. *Cartography and Geographic Information Science*. Taylor & Francis, 2023, vol. 50, p. 385–402. ISSN 1545-0465.

[59] Gedicke, S.; Jabrayilov, A.; Niedermann, B.; Mutzel, P. and Haunert, J. H. Point feature label placement for multi-page maps on small-screen devices. *Computers and Graphics (Pergamon)*. Elsevier Ltd, 2021, vol. 100, p. 66–80. ISSN 0097-8493.

[60] Gemsa, A.; Haunertand, J.-H. and Nöllenburg, M. Multi-Row Boundary-Labeling Algorithms for Panorama Images. *ACM TSAS*, 2014, vol. 1, no. 1, p. 289–298. ISSN 2374-0353.

[61] Gemsa, A.; Niedermann, B. and Nöllenburg, M. Trajectory-Based Dynamic Map Labeling. In: *Algorithms and Computation*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, p. 413–423. ISBN 978-3-642-45030-3.

[62] Gemsa, A.; Niedermann, B. and Nöllenburg, M. A Unified Model and Algorithms for Temporal Map Labeling. *Algorithmica*. Springer US, 2020. ISSN 1432-0541.

[63] Gemsa, A.; Nöllenburg, M. and Rutter, I. Consistent labeling of rotating maps. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2011, 6844 LNCS, p. 451–462. ISSN 0302-9743.

[64] Gemsa, A.; Nöllenburg, M. and Rutter, I. Evaluation of Labeling Strategies for Rotating Maps. In: *Experimental Algorithms*. Cham: Springer International Publishing, 2014, p. 235–246. ISBN 978-3-319-07959-2.

[65] Götzelmann, T.; Hartmann, K. and Strothotte, T. Agent-based annotation of interactive 3D visualizations. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, 4073 LNCS, p. 24–35. ISSN 1611-3349.

[66] Götzelmann, T.; Hartmann, K. and Strothotte, T. Annotation of animated 3D objects. In: *SimVis, SCS*. Publishing House, 2007, p. 209–222.

[67] Gurobi Optimization, L. *Advanced Gurobi Algorithms*. 2016. Available at: http://www.gurobi.com/pdfs/user-events/2016-frankfurt/Die-Algorithmen.pdf. Accessed: 2024-06-05.

[68] Harrie, L.; Oucheikh, R.; Nilsson Åsa; Oxenstierna, A.; Cederholm, P. et al. Label Placement Challenges in City Wayfinding Map Production—Identification and Possible Solutions. *Journal of Geovisualization and Spatial Analysis*. Springer International Publishing, 2022, vol. 6. ISSN 2509-8810.

[69] Hartmann, K.; Ali, K. and Strothotte, T. Floating Labels: Applying Dynamic Potential Fields for Label Layout. In: *Smart Graphics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 101–113. ISBN 978-3-540-24678-7.

[70] Hartmann, K.; Götzelmann, T.; Ali, K. and Strothotte, T. Metrics for Functional and Aesthetic Label Layouts. In: Butz, A.; Fisher, B.; Krüger, A. and Olivier, P., ed. *Smart Graphics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, p. 115–126. ISBN 978-3-540-31905-4.

[71] Hastie, T.; Tibshirani, R. and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer, 2009. Springer series in statistics. ISBN 978-0-387-84884-6.

[72] Haunert, J.-H. and Wolff, A. Beyond Maximum Independent Set: An Extended Integer Programming Formulation for Point Labeling. *ISPRS International Journal*

of Geo-Information, 2017, vol. 6, no. 11. ISSN 2220-9964.

[73] He, Y.; Zhao, G. D. and Zhang, S. H. Smoothness preserving layout for dynamic labels by hybrid optimization. *Computational Visual Media*, 2022, vol. 8, p. 149–163. ISSN 2096-0662.

[74] Hegde, S.; Maurya, J.; Hebbalaguppe, R. and Kalkar, A. SmartOverlays: A visual saliency driven label placement for intelligent human-computer interfaces. *Proceedings - 2020 IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, 2020, p. 1110–1119. ISSN 2642-9381.

[75] Heinsohn, N.; Gerasch, A. and Kaufmann, M. Boundary labeling methods for dynamic focus regions. *IEEE Pacific Visualization Symposium*, 2014, p. 243–247. ISSN 2165-8773.

[76] Higashikawa, Y.; Imai, K.; Shiraga, T.; Sukegawa, N. and Yokosuka, Y. Minimum point-overlap labelling*. *Optimization Methods and Software*, 2021, vol. 36, p. 316–325. ISSN 1029-4937.

[77] Hirsch, S. A. An Algorithm for Automatic Name Placement Around Point Data. *The American Cartographer*. Taylor & Francis, 1982, vol. 9, no. 1, p. 5–17. ISSN 2330-9164.

[78] Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*. [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley], 1979, vol. 6, no. 2, p. 65–70. ISSN 0303-6898.

[79] Hu, R.; Chen, B.; Xu, J.; Kaick, O. van; Deussen, O. et al. Shape-driven Coordinate Ordering for Star Glyph Sets via Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics*, march 2021, vol. 27, p. 3034–3047. ISSN 1941-0506.

[80] Hua Liao, W. D. and Meng, L. Measuring the influence of map label density on perceived complexity: a user study using eye tracking. *Cartography and Geographic Information Science*. Taylor & Francis, 2019, vol. 46, no. 3, p. 210–227. ISSN 1545-0465.

[81] Huang, Z.-D.; Poon, S.-H. and Lin, C.-C. Boundary Labeling with Flexible Label Positions. In: *Algorithms and Computation*. Springer, 2014, p. 44–55. ISBN 978-3-319-04656-3.

[82] Imhof, E. *Positioning names on maps*. 1975. ISSN 0094-1689.

[83] Jain, A. K.; Murty, M. N. and Flynn, P. J. Data clustering: a review. *ACM Comput. Surv.* New York, NY, USA: Association for Computing Machinery, sep 1999, vol. 31, no. 3, p. 264–323. ISSN 0360-0300.

[84] Jaunet, T.; Vuillemot, R. and Wolf, C. DRLViz: Understanding Decisions and Memory in Deep Reinforcement Learning. *Computer Graphics Forum*, 2020, vol. 39, no. 3, p. 49–61. ISSN 1467-8659.

[85] Jia, J.; Elezovikj, S.; Fan, H.; Yang, S.; Liu, J. et al. Semantic-aware label placement for augmented reality in street view. *The Visual Computer*, 2021, vol. 37, no. 7, p. 1805–1819. ISSN 0178-2789.

[86] Jones, C. B. Cartographic Name Placement with Prolog. *IEEE Computer Graphics and Applications*, 1989, vol. 9, p. 36–47. ISSN 0272-1716.

[87] Kakoulis, K. G. and Tollis, I. G. A unified approach to labeling graphical features. In: *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*. New York, NY, USA: Association for Computing Machinery, 1998, p. 347–356. SCG '98. ISBN 978-0-89791-973-9.

[88] Kato, T. and Imai, H. The NP-completeness of the character placement problem of

2 or 3 degrees of freedom. In: *Record of Joint Conference of Electrical and Electronic Engineers in Kyushu*. 1988, vol. 1138.

[89] Kendall, M. G. and Smith, B. On the method of paired comparisons. *Biometrika*, 1947, vol. 34, p. 363–365. ISSN 0006-3444.

[90] Kindermann, P.; Lipp, F. and Wolff, A. Luatodonotes: Boundary Labeling for Annotations in Texts. In: *Graph Drawing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, p. 76–88. ISBN 978-3-662-45803-7.

[91] Kindermann, P.; Niedermann, B.; Rutter, I.; Schaefer, M.; Schulz, A. et al. Multi-sided Boundary Labeling. *Algorithmica*. Springer US, 2016, vol. 76, p. 225–258. ISSN 1432-0541.

[92] Kittivorawong, C.; Moritz, D.; Wongsuphasawat, K. and Heer, J. Fast and Flexible Overlap Detection for Chart Labeling with Occupancy Bitmap. In: *2020 IEEE Visualization Conference (VIS)*. 2020, p. 101–105. ISBN 978-1-7281-8014-4/20.

[93] Kittivorawong, C. *Legible Label Layout for Data Visualization, Algorithm and Integration into Vega-Lite*. 2024. Available at: https://doi.org/10.48550/arXiv.2405.10953.

[94] Klau, G. W. and Mutzel, P. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming*, 2003, vol. 94, p. 435–458. ISSN 0025-5610.

[95] Klau, G. W. A Combinatorial Approach to Orthogonal Placement Problems. In: *Operations Research Proceedings 2002*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, p. 26–32. ISBN 978-3-642-55537-4.

[96] Klute, F.; Li, G.; Löffler, R.; Nöllenburg, M. and Schmidt, M. Exploring Semi-Automatic Map Labeling. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery, 2019, p. 13–22. SIGSPATIAL '19. ISBN 978-1-4503-6909-1.

[97] Kouřil, D.; Čmolík, L.; Kozlíková, B.; Wu, H.; Johnson, G. et al. Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments. *IEEE Transactions on Visualization and Computer Graphics*, 2019, vol. 25, no. 1, p. 977–986. ISSN 1077-2626.

[98] Krygier, J. and Wood, D. *Making maps: a visual guide to map design for GIS*. 3rd ed. Guilford Publications, 2016. ISBN 978-1-4625-0998-0.

[99] Lamm, S.; Sanders, P.; Schulz, C.; Strash, D. and Werneck, R. F. Finding near-optimal independent sets at scale. *Journal of Heuristics*, 2017, vol. 23, no. 4, p. 207–229. ISSN 1572-9397.

[100] Le Muzic, M.; Autin, L.; Parulek, J. and Viola, I. CellVIEW: A Tool for Illustrative and Multi-Scale Rendering of Large Biomolecular Datasets. In: *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*. Goslar, DEU: Eurographics Association, 2015, p. 61–70. VCBM '15. ISBN 978-3-90567482-8.

[101] Lehtinen, V. *Automatic Label Placement for Technical Drawings*. 2020. Master's thesis. University of Turku. Available at: https://www.utupub.fi/handle/10024/150279. Accessed: 2024-08-05.

[102] Lessani, M. N.; Li, Z.; Deng, J. and Guo, Z. An MPI-based parallel genetic algorithm for multiple geographical feature label placement based on the hybrid of fixed-sliding models. *Geo-Spatial Information Science*. Taylor & Francis, 2024,

vol. 00, p. 1–19. ISSN 1009-5020.

[103] Levine, S.; Finn, C.; Darrell, T. and Abbeel, P. End-to-End Training of Deep Visuomotor Policies. *The Journal of Machine Learning Research*. JMLR.org, 2016, vol. 17, no. 1, p. 1334–1373. ISSN 1532-4435.

[104] Levine, S.; Pastor, P.; Krizhevsky, A. and Quillen, D. Learning Hand-Eye Coordination for Robotic Grasping with Large-Scale Data Collection. In: *2016 International Symposium on Experimental Robotics*. Cham: Springer International Publishing, 2017, vol. 1. ISSN 0278-3649.

[105] Lhuillier, A.; Garderen, M. van and Weiskopf, D. Density-based label placement. *Visual Computer*. Springer Berlin Heidelberg, 2019, vol. 35, no. 6, p. 1041–1052. ISSN 0178-2789.

[106] Li, K. and Malik, J. Learning to optimize. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. 2017, p. 1–13. ISBN 978-1-71387-271-9.

[107] Li, L.; Zhang, H.; Zhu, H.; Kuai, X. and Hu, W. A labeling model based on the region of movability for point-feature label placement. *ISPRS International Journal of Geo-Information*. MDPI AG, sep 2016, vol. 5, no. 9. ISSN 2220-9964.

[108] Liang, E.; Liaw, R.; Nishihara, R.; Moritz, P.; Fox, R. et al. RLlib: Abstractions for Distributed Reinforcement Learning. In: *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018, vol. 80, p. 3059–3068. ISBN 978-1-5108-6796-3.

[109] Liao, C. S.; Liang, C. W. and Poon, S. H. Approximation algorithms on consistent dynamic map labeling. *Theoretical Computer Science*, 2016, vol. 640. ISSN 0304-3975.

[110] Lin, C. C. Crossing-free many-to-one boundary labeling with hyperleaders. In: *IEEE Pacific Visualization Symposium 2010, PacificVis 2010 - Proceedings*. IEEE, 2010, vol. 2, p. 185–192. ISBN 978-1-4244-6684-9.

[111] Lin, C. C.; Kao, H. J. and Yen, H. C. Many-to-one boundary labeling. In: *Asia-Pacific Symposium on Visualisation 2007, APVIS 2007, Proceedings*. 2007, vol. 12, no. 3, p. 65–72. ISBN 978-1-4244-0808-5.

[112] Lin, C. C.; Wu, H. Y. and Yen, H. C. Boundary labeling in text annotation. *Proceedings of the International Conference on Information Visualisation*, 2009, p. 110–115. ISSN 1093-9547.

[113] Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P. et al. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017, p. 6382–6393. ISBN 978-1-5108-6096-4.

[114] Lu, F.; Deng, J.; Li, S. and Deng, H. A hybrid of differential evolution and genetic algorithm for the multiple geographical feature label placement problem. *ISPRS International Journal of Geo-Information*, 2019, vol. 8, no. 5. ISSN 2220-9964.

[115] Luboschik, M.; Schumann, H. and Cords, H. Particle-based labeling: Fast point-feature labeling without obscuring other visual features. *IEEE Transactions on Visualization and Computer Graphics*, 2008, vol. 14, no. 6, p. 1237–1244. ISSN 1077-2626.

[116] Ludbrook, J. Multiple comparison procedures updated. *Clinical and Experimental Pharmacology and Physiology*, 1998, vol. 25, no. 12, p. 1032–1037. ISSN 0305-1870.

[117] Maass, S. and Döllner, J. Efficient View Management for Dynamic Annotation

Placement in Virtual Landscapes. In: *Smart Graphics*. Springer, 2006, p. 1–12. ISBN 978-3-540-36295-1.

[118] MacKenzie, I. S. *Human-computer interaction: An empirical research perspective*. 1st ed. Morgan Kaufmann Publishers Inc., 2013. ISBN 978-0-12-405865-1.

[119] Madsen, J. B.; Tatzqern, M.; Madsen, C. B.; Schmalstieg, D. and Kalkofen, D. Temporal Coherence Strategies for Augmented Reality Labeling. *IEEE Transactions on Visualization and Computer Graphics*, 2016. ISSN 1077-2626.

[120] Marc Van Kreveld, A.; Tycho Strijk, A. and Er Wolff, B. Point Labeling with Sliding Labels. *Computational Geometry: Theory and Applications*, 1999, vol. 13, Special issue on applications and challenges, p. 21–47. ISSN 0925-7721.

[121] Marks, J. and Shieber, S. M. *The Computational Complexity of Cartographic Label Placement*. Center for Research in Computing Technology, Harvard University, 1991.

[122] Marín, A. and Pelegrín, M. Towards unambiguous map labeling - Integer programming approach and heuristic algorithm. *Expert Systems with Applications*, 2018, vol. 98, p. 221–241. ISSN 0957-4174.

[123] Mauri, G. R.; Ribeiro, G. M. and Lorena, L. A. A new mathematical model and a Lagrangean decomposition for the point-feature cartographic label placement problem. *Computers and Operations Research*. Elsevier, 2010, vol. 37, p. 2164–2172. ISSN 0305-0548.

[124] Mills, W. and Buttenfield, B. P. Practical considerations in name placement: a defence of Pinhas Yoeli (and response). *Cartographica*, Winter 1994 1994, vol. 31, no. 4, p. 58–62. ISSN 0317-7173.

[125] Mirhoseini, A.; Goldie, A.; Yazgan, M.; Jiang, J. W.; Songhori, E. et al. A graph placement methodology for fast chip design. *Nature*. Springer US, 2021, vol. 594, no. 7862, p. 207–212. ISSN 0028-0836.

[126] Mishra, A.; Soni, U.; Huang, J. and Bryan, C. Why? Why not? When? Visual Explanations of Agent Behaviour in Reinforcement Learning. In: *2022 IEEE 15th Pacific Visualization Symposium (PacificVis)*. Los Alamitos, CA, USA: IEEE Computer Society, Apr 2022, p. 111–120. ISBN 978-1-6654-2335-9.

[127] Mogalle, K.; Tietjen, C.; Soza, G. and Preim, B. Constrained Labeling of 2D Slice Data for Reading Images in Radiology. In: Ropinski, T.; Ynnerman, A.; Botha, C. and Roerdink, J., ed. *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2012. ISBN 978-3-905674-38-5.

[128] Mote, K. Fast point-feature label placement for dynamic visualizations. *Information Visualization*. Palgrave Macmillan, dec 2007, vol. 6, no. 4, p. 249–260. ISSN 1473-8716.

[129] Mühler, K. and Preim, B. Automatic textual annotation for surgical planning. In: *VMV '09*. 2009, p. 277–284. ISBN 978-3-98048748-1.

[130] Nascimento, H. A. D.; Eades, P.; Nascimento, H. A. D. do and Eades, P. User Hints for map labeling. *Journal of Visual Languages and Computing*, 2008, vol. 19, no. 1, p. 39–74. ISSN 1045-926X.

[131] Nguyen, D. T.; Kumar, A. and Lau, H. C. Credit Assignment for Collective Multiagent RL with Global Rewards. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 8113–8124. ISBN 978-1-5108-8447-2.

[132] Niedermann, B. and Haunert, J. H. Focus+context map labeling with optimized

clutter reduction. *International Journal of Cartography*. Taylor and Francis Ltd., may 2019, vol. 5, p. 158–177. ISSN 2372-9341.

[133] Niedermann, B.; Nollenburg, M. and Rutter, I. Radial contour labeling with straight leaders. *IEEE Pacific Visualization Symposium*, 2017, p. 295–304. ISSN 2165-8773.

[134] Oeltze Jafra, S. and Preim, B. Survey of Labeling Techniques in Medical Visualizations. In: *Eurographics Workshop on Visual Computing for Biology and Medicine*. The Eurographics Association, 2014, p. 199–208. ISBN 978-3-905674-62-0.

[135] Ogus, S. Augmented Reality Options by Second Spectrum Added to ESPN App for NBA Playoffs. *Forbes*, May 2019. Available at: https://www.forbes.com/sites/simonogus/2019/05/26/augmented-reality-options-by-second-spectrum-added-to-espn-app-for-nba-playoffs/. Accessed: 2024-08-13.

[136] Pan, X.; You, Y.; Wang, Z. and Lu, C. Virtual to Real Reinforcement Learning for Autonomous Driving. In: *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, 2017, p. 11.1–11.13. ISBN 1-901725-60-X.

[137] Paulsen, F. and Waschke, J. *Sobotta Atlas of Human Anatomy, Vol.3, 16th ed., English: Head, Neck and Neuroanatomy*. 16th ed. Elsevier, 2018. ISBN 978-0-7020-5271-2.

[138] Pavlovec, V. and Čmolík, L. Rapid Labels: Point-Feature Labeling on GPU. *IEEE Transactions on Visualization and Computer Graphics*, 2022, vol. 28, no. 1, p. 604–613. ISSN 1941-0506.

[139] Pérez Ortiz, M. and Mantiuk, R. K. *A practical guide and software for analysing pairwise comparison experiments*. 2017.

[140] Petzold, I.; Gröger, G. and Plümer, L. Fast screen map labeling – data-structures and algorithms. In: *Proceedings of the 21st International Cartographic Conference*. 2003, p. 426–437. ISBN 978-0-9584609-3-4.

[141] Preim, B.; Ritter, A.; Strothotte, T.; Pohle, T.; Bartram, L. et al. Consistency of Rendered Images and Their Textual Label. In: *Proceedings of Compugraphics´95*. 1995, p. 201–210.

[142] Purchase, H. Which aesthetic has the greatest effect on human understanding? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1997, vol. 1353, p. 248–261. ISSN 1611-3349.

[143] Rabello, R. L.; Mauri, G. R.; Ribeiro, G. M. and Lorena, L. A. N. A Clustering Search metaheuristic for the Point-Feature Cartographic Label Placement Problem. *European Journal of Operational Research*, 2014, vol. 234, no. 3, p. 802–808. ISSN 0377-2217.

[144] Ribeiro, G. M. and Lorena, L. A. N. Column generation approach for the point-feature cartographic label placement problem. *Journal of Combinatorial Optimization*, february 2008, vol. 15, p. 147–164. ISSN 1382-6905.

[145] Ribeiro, G. M. and Lorena, L. A. N. Lagrangean relaxation with clusters for point-feature cartographic label placement problems. *Computers and Operations Research*, july 2008, vol. 35, p. 2129–2140. ISSN 0305-0548.

[146] Robinson, A. H.; Morrison, J.; Muehrcke, P.; Kimerling, A. and Guptill, S. *Elements of Cartography*. 6th ed. Wiley, 1995. ISBN 978-0-471-55579-7.

[147] Robinson, A. H.; Morrison, J. L. and Muehrcke, P. C. Cartography 1950-2000. *Transactions of the Institute of British Geographers*. [Royal Geographical Society (with the Institute of British Geographers), Wiley], 1977, vol. 2, no. 1, p. 3–18.

ISSN 0020-2754.
[148] Rylov, M. A. and Reimer, A. W. A Comprehensive Multi-criteria Model for High Cartographic Quality Point-Feature Label Placement. *Cartographica*, 2014, vol. 49, no. 1, p. 52–68. ISSN 0317-7173.
[149] Sauro, J. and Lewis, J. R. Estimating completion rates from small samples using binomial confidence intervals: comparisons and recommendations. In: *Proceedings of the human factors and ergonomics society annual meeting*. 2005, 49, no. 24, p. 2100–2103. ISSN 2169-5067.
[150] Sauro, J. and Lewis, J. R. *Quantifying the user experience: Practical statistics for user research*. 1st ed. Elsevier, 2012. ISBN 978-0-12-384968-7.
[151] Scheuerman, J.; Harman, J. L.; Goldstein, R. R.; Acklin, D. and Michael, C. J. Visual preferences in map label placement. *Discover Psychology*, 2023, vol. 3, p. 27. ISSN 2731-4537.
[152] Schubert, S.; Neubert, P.; Pöschmann, J. and Protzel, P. Circular Convolutional Neural Networks for Panoramic Images and Laser Data. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019, p. 653–660. ISBN 978-1-72810-561-1.
[153] Schulman, J.; Moritz, P.; Levine, S.; Jordan, M. I. and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2016.
[154] Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A. and Klimov, O. Proximal policy optimization algorithms. *CoRR*, 2017.
[155] Šidák, Z. Rectangular Confidence Regions for the Means of Multivariate Normal Distributions. *Journal of the American Statistical Association*. Taylor & Francis, 1967, vol. 62, no. 318, p. 626–633. ISSN 0162-1459.
[156] Singh, S.; Kumar, V. and Bhattacharjee, S. An Integrated Approach to Point Labeling Problem. In: *14th International Conference on Computing Communication and Networking Technologies, ICCCNT 2023*. IEEE, 2023, p. 1–6. ISBN 979-835033509-5.
[157] Sirk, C.; Kalkofen, D.; Schmalstieg, D. and Bornik, A. Dynamic Label Placement for Forensic Case Visualization. In: *EG/VGTC Conference on Visualization (Eurovis'17)*. 2017, p. 139–143. ISBN 978-3-03868-043-7.
[158] Slocum, T. A.; McMaster, R. B.; Kessler, F. C. and Howard, H. H. *Thematic Cartography and Geovisualization*. 3rd ed. Pearson, 2009. ISBN 978-0-13-229834-6.
[159] Stein, T. and Décoret, X. Dynamic Label Placement for Improved Interactive Exploration. *Proc. of NPAR '08*, 2008, p. 15–21.
[160] Steven Van Dijk, T. S. and Wolff, A. Towards an evaluation of quality for names placement methods. *International Journal of Geographical Information Science*. Taylor & Francis, 2002, vol. 16, no. 7, p. 641–661. ISSN 1365-8816.
[161] Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. 2nd ed. MIT press, 2018. ISBN 978-0-262-03924-6.
[162] Tang, T.; Li, R.; Wu, X.; Liu, S.; Knittel, J. et al. PlotThread: Creating expressive storyline visualizations using reinforcement learning. *IEEE Transactions on Visualization and Computer Graphics*, 2021, vol. 27, no. 2, p. 294–303. ISSN 1941-0506.
[163] Tatzgern, M.; Kalkofen, D.; Grasset, R. and Schmalstieg, D. Hedgehog labeling: View management techniques for external labels in 3D space. In: *2014*

IEEE Virtual Reality. 2014, p. 27–32. ISSN 1087-8270.

[164] Tech, A. S. *Aircraft Radar Beacon Transponder*. 2017. Available at: https://www.aircraftsystemstech.com/2017/05/radar-beacon-transponder.html. Accessed: 2024-08-06.

[165] Thurstone, L. L. A law of comparative judgment. *Psychological Review*, 1927, vol. 34, p. 273–286. ISSN 0033-295X.

[166] Tsukida, K. and Gupta, M. R. How to Analyze Paired Comparison Data. *UWEE Technical Report 206*, 2011.

[167] Tyner, J. A. *Principles of map design*. 1st ed. Guilford Publications, 2014. ISBN 978-1-4625-1712-1.

[168] Vaaraniemi, M.; Treib, M. and Westermann, R. Temporally Coherent Real-time Labeling of Dynamic Scenes. In: *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*. New York, NY, USA: ACM, 2012, p. 17:1–17:10. COM.Geo '12. ISBN 978-1-4503-1113-7.

[169] Verweij, B. and Aardal, K. An Optimisation Algorithm for Maximum Independent Set with Applications in Map Labelling. In: *Algorithms - ESA' 99*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, p. 426–437. ISBN 978-3-540-48481-3.

[170] Wagner, F. and Wolff, A. Map labeling heuristics : provably good and practically useful. In: *Proceedings 11th Annual ACM Symposium on Computational Geometry (Vancouver BC, Canada, June 5-12, 1995)*. United States: Association for Computing Machinery, Inc, 1995, p. 109–118. ISBN 0-89791-724-3.

[171] Wagner, F.; Wolff, A.; Kapoor, V. and Strijk, T. Three Rules Suffice for Good Label Placement. *Algorithmica*. Berlin, Heidelberg: Springer-Verlag, jun 2001, vol. 30, no. 2, p. 334–349. ISSN 0178-4617.

[172] Wagner, F. Approximate map labeling is in $\Sigma(n \log n)$. *Information Processing Letters*, 1994, vol. 52, no. 3, p. 161–165. ISSN 0020-0190.

[173] Wang, J.; Gou, L.; Shen, H.-W. and Yang, H. DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks. *IEEE Transactions on Visualization and Computer Graphics*, 2018, vol. 25, no. 1, p. 288–298. ISSN 1941-0506.

[174] Wang, J.; Zhang, W.; Yang, H.; Yeh, C.-C. M. and Wang, L. Visual Analytics for RNN-Based Deep Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics*, 2022, vol. 28, no. 12, p. 4141–4155. ISSN 1941-0506.

[175] Wang, Q.; Chen, Z.; Wang, Y. and Qu, H. A Survey on ML4VIS: Applying Machine Learning Advances to Data Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 2022, vol. 28, no. 12, p. 5134–5153. ISSN 1941-0506.

[176] Ware, C.; Purchase, H.; Colpoys, L. and McGill, M. Cognitive Measurements of Graph Aesthetics. *Information Visualization*, 2002, vol. 1, no. 2, p. 103–110. ISSN 1473-8716.

[177] Wickelmaier, F. and Schmid, C. A Matlab function to estimate choice model parameters from paired-comparison data. *Behavior Research Methods, Instruments, and Computers*, 2004, vol. 36, no. 1, p. 29–40. ISSN 1532-5970.

[178] Wood, C. H. A descriptive and illustrated guide for type placement on small scale maps. *Cartographic Journal*, 2000, vol. 37, p. 5–18. ISSN 0008-7041.

[179] Wu, A.; Tong, W.; Dwyer, T.; Lee, B.; Isenberg, P. et al. MobileVisFixer: Tailoring Web Visualizations for Mobile Phones Leveraging an Explainable Reinforcement Learning Framework. *IEEE Transactions on Visualization and*

Computer Graphics, 2021, vol. 27, no. 2, p. 464–474. ISSN 1941-0506.

[180] Wu, C. V. and Buttenfield, B. P. Reconsidering rules for point-feature name placement. *Cartographica*, 1991, vol. 28, no. 1, p. 10–27. ISSN 0317-7173.

[181] Yamamoto, D.; Ozeki, S. and Takahashi, N. Focus+Glue+Context: an improved fisheye approach for web map services. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. New York, NY, USA: Association for Computing Machinery, 2009, p. 101–110. GIS '09. ISBN 978-1-60558-649-6.

[182] Yamamoto, M.; Camara, G. and Lorena, L. A. N. Fast point-feature label placement algorithm for real time screen maps. In: *GEOINFO 2005 - 7th Brazilian Symposium on GeoInformatics*. 2005. ISBN 978-851700022-5.

[183] Yamamoto, M.; Camara, G. and Nogueira Lorena, L. A. Tabu search heuristic for point-feature cartographic label placement. *GeoInformatica*, 2002, vol. 6, no. 1, p. 77–90. ISSN 1384-6175.

[184] Ye, Y. and Tse, E. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, 1989, vol. 44, no. 1, p. 157–179. ISSN 1436-4646.

[185] Yoeli, P. The Logic of Automated Map Lettering. *The Cartographic Journal*. Taylor & Francis, 1972, vol. 9, no. 2, p. 99–108. ISSN 1743-2774.

[186] Yokosuka, Y. and Imai, K. Polynomial time algorithms for label size maximization on rotating maps. *Journal of Information Processing*, 2013, vol. 25, p. 572–579. ISSN 1882-6652.

[187] Zhang, X.; Poon, S.-h.; Li, M. and Lee, V. C. S. On Maxmin Active Range Problem for Weighted Consistent Dynamic Map Labeling. In: *GEOProcessing*. 2015, vol. 1, c, p. 32–37. ISBN 978-1-61208-383-4.

[188] Zhang, X.; Poon, S.-H.; Liu, S.; Li, M. and Lee, V. C. Consistent Dynamic Map Labeling with Fairness and Importance. *Computer Aided Geometric Design*. Elsevier B.V., 2020, vol. 81, p. 101892. ISSN 0167-8396.

[189] Zhou, M.; Li, Q.; He, X.; Li, Y.; Liu, Y. et al. Table2Charts: Recommending Charts by Learning Shared Table Representations. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2021, p. 2389–2399. KDD '21. ISBN 978-1-4503-8332-5.

[190] Zoraster, S. Integer programming applied to the map label placement problem. *Cartographica*, 1986, vol. 23, p. 16–27. ISSN 0317-7173.

[191] Zoraster, S. Integer Programming Applied To The Map Label Placement Problem. *Cartographica*, 1986, vol. 23, no. 3, p. 16–27. ISSN 0317-7173.

[192] Zoraster, S. Solution of large 0-1 integer programming problems encountered in automated cartography. *Operations Research*, 1990, vol. 38, p. 752–759. ISSN 0030-364X.

[193] Zoraster, S. Practical Results Using Simulated Annealing for Point Feature Label Placement. *Cartography and Geographic Information Systems*. Taylor & Francis, 1997, vol. 24, no. 4, p. 228–238. ISSN 2330-9172.

Appendix A

Temporally Stable Optimization Approach to Boundary Labeling

In this appendix, we present a detailed description of the tasks that participants fulfilled during the user study. These tasks were designed to evaluate the accuracy and preference of labeling methods in both non-interactive and interactive environments. This additional information complements the findings discussed in the main document, providing a deeper understanding of the user study and its outcomes.

A.1 Accuracy Experiments

In this section, we present a detailed description of the tasks evaluated in the Accuracy Experiments. To see all the three described tasks being fulfilled, please look at the supplementary video.[1] Please note that the participants tried each task on a demo scene to get familiar with the task before they started fulfilling the task.

A.1.1 Task 1: Assign Label to Highlighted Anchor

The first task was to find the label associated with a highlighted anchor. An image with a labeled scene was presented to the participant. After one second, one of the anchors was highlighted. The participant had to find the label associated with the highlighted anchor and press the space-bar. Then only the labels (without text) remained visible on the screen, and the participant had to click on the associated label. The participant was instructed to click on free space between labels if s/he was not able to find the associated label. We measured the reaction *time*, measured as the time between highlighting the anchor and pressing the space-bar, and the *error rate*, measured as the number of wrongly selected labels relative to all selected labels. When the participant could not decide which label belonged to the highlighted anchor, we counted this as an error. The whole process was repeated 30 times (10 different labels in 3 different scenes). The participants indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5. After each scene, we have presented to the participant three statements:

(**1**) It was easy to select the correct labels.

(**2**) I was able to find the correct labels fast.

[1] Supplementary video is available at the project page http://cphoto.fit.vutbr.cz/interactive-labeling/

(**3**) I was confident that I was selecting the correct labels.

A.1.2 Task 2: Assign Anchor to Highlighted Label

The second task was to find the anchor associated with a highlighted label. The task was similar to task 1. An image with a labeled scene was presented to the participant. After one second, one of the labels was highlighted. The participant had to find the anchor associated with the highlighted label and press the space-bar. Then only small boxes around each anchor remained visible on the screen, and the participant had to click on the associated anchor. The participant was instructed to click on free space between the boxes if s/he was not able to find the associated anchor. Similarly, as in task 1 the reaction *time* and the *error rate*. The whole process was repeated 30 times (10 different labels in 3 different scenes). The participants indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5. After each scene, we have presented to the participant three statements:

(**1**) It was easy to select the correct anchors.
(**2**) I was able to find the correct anchors fast.
(**3**) I was confident that I was selecting the correct anchors.

A.1.3 Task 3: Follow the Moving Label

The third task was to follow a certain label moving in time and then select the label. The task was similar to tasks 1 and 2. An image with a labeled scene was presented to the participant. After one second, one of the labels was highlighted. The participant pressed the space-bar, then the highlight of the label disappeared, and the animation started playing with the speed of 10 frames per second. The participant had to follow the movement of the initially highlighted label. After two seconds, the animation stopped, and only the labels (without text) were displayed on the screen. The participant should have clicked the label that s/he was following. The participant was instructed to click on free space between the boxes if s/he was not able to find the correct label. In this task, we measured the *error rate* only. Again, the whole process was repeated 30 times (10 different labels in 3 different scenes). The participants indicated their subjective agreement or disagreement with each statement on Likert scale from 1 to 5. After each scene, we have presented to the participant two statements:

(**1**) The label layout made it easy to follow the labels.
(**2**) I didn't have to focus hard to be able to follow the labels.

A.2 Preference Experiments

In this section, we present the detailed description of the tasks evaluated in non-interactive and interactive environments. To see all the three described tasks being fulfilled, please look at the supplementary video[1].

A.2.1 Non-Interactive Environment

The stimuli were represented by three different video sequences presented in a web browser, and we evenly distributed them among the participants. Each participant was sequentially stimulated by a pair of videos produced by the tested labeling methods. At the beginning

of the experiment, participants were familiarized with the experimental procedure by the written instructions:

(**1**) When the test starts, you will be able to play two animations, please play both of them.

(**2**) During each playback, try to focus on your personal feeling about the visual presentation of labels (label layout).

(**3**) Afterward, you will be asked if you prefer the first or second label layout.

During the experiment, participants were able to play the assigned sequence as many times as they wanted. The names of the methods were transcoded with numbers.

A.2.2 Interactive Environment

The stimuli were represented by an interactive visualization of mountain terrain presented at resolution 1200x900, where the independent variable was the labeling method. During the experiment, participants were guided along the same predefined path above mountain peaks. Each participant was familiarized with the experimental procedure by the written instructions:

(**1**) This study aims to assess four interactive labeling techniques for dynamic scenes. The estimated time to complete the study is approximately 20 minutes.

(**2**) The presented scene is set in the Alps where you can:

- (**a**) fly above mountain tops following the predefined path by holding the Up key (forward) or the Down key (backward),
- (**b**) stop whenever you want to by lifting your finger from the Up or Down key,
- (**c**) rotate the camera by pressing the left mouse button and moving,
- (**d**) zoom in and out by mouse wheel, and
- (**e**) return to the beginning of the path by pressing the Z key.

(**3**) Your goal is to rank the four methods „1“, „2“, „3“, „4“ from the most preferred to the least preferred method. You can use prepared cards to establish the ranking continuously during the study.

(**4**) Please focus predominantly on the assessment of the label placement and the movement of the labels in the time.

(**5**) Throughout the study, you can change the method any time by pressing the corresponding numerical key. The name of the selected method is available in the top left corner of the application window. You can change the method as many times as you want.

(**6**) Please let the supervisor know whenever you are satisfied with the final ranking of the methods or when you need assistance.

At the end of the experiment, participants were asked to describe their decision process and to justify their ranking.

Appendix B

Deep Reinforcement Learning Approach to Internal Labeling

In this appendix, we provide a collection of additional resources, including dataset definitions, experimental setups, supplementary results, and visual comparisons that complement the main findings discussed in the manuscript. These elements were omitted from the main document in an effort to preserve its conciseness and readability. Our aim is to ensure that readers have access to all necessary supplementary information to fully understand the scope and depth of our research findings.

B.1 Hyperparameters

We use PrPO implementation from the RLLib framework [108]. Rollout workers query the current policy to determine actions and collect a new vector of observations and rewards. Collected data are assembled into training batches and shuffled. The trainer worker coordinates the rollout workers and orchestrates policy optimization. We use Adam to optimize the policy and value function parameters. Further hyperparameters are as follows:

Parameter	Value
Learning rate	$1 \cdot 10^{-6}$
Batch size	2000
Mini-batch size	128
SGD epochs	10
PrPO clip factor	0.25
PrPO gradient clipping	–
PrPO entropy coefficient	0.0
Horizon	100
γ	0.99
λ	1.0

Table B.1: Hyperparameters

B.2 Observation Modalities

We designed two observation vectors. First, the *mapping* vector M provides surroundings modalities. Second, the *self-aware* vector S provides local information about the state of the given label agent. Observations are mostly normalized to $[-1, 1]$ range. The intersection

Mapping vector	Range	Shape
Intersection distance	$(-1, 1)$	32
Intersection mass	$(-1, 1)$	32
Number of intersected labels	$(-1, 1)$	32
Intersection object type	$(-1, 0, 1)$	32
Self-aware vector	Range	Shape
Overlap area	$(-1, 1)$	1
Overlap indicator	$(-1, 1)$	1
Number of overlaps	$(-1, 1)$	1
Displacement	$(-1, 1)$	1
Cumulative displacement	$(-1, 1)$	1
Anchor penetration distance	$(-1, 1)$	1
Penetration indicator	$(-1, 1)$	1
Number of penetrations	$(-1, 1)$	1
Anchor-port distance	$(-1, 1)$	1
Anchor-port angle	$(-1, 1)$	1
Anchor-port vector	$(-1, 1)$	2
Anchor-origin distance	$(-1, 1)$	1
Elapsed time steps	$(-1, 1)$	1

Table B.2: Observation Modalities

observation type is an exception – the value of -1 signifies the intersection of anchor, 0 defines screen bound, and 1 represents the label. We use Box2D[1] framework to cast rays serving as LiDAR sensor to encode the agent's surroundings by 32 rays (similar to LiDAR) evenly distributed around the label bounds that sense the range, type of the nearest intersected object (*i.e.*, label, anchor, bounds of the environment), and the mass that the ray went through.

B.3 Dataset Definition

The dataset comprises a collection of instances, where a JSON file represents each instance. The JSON file contains information about the screen or drawing area, including the width and height, as well as an optional reference to a background image. Additionally, the JSON file contains a description of the labels associated with the instance, where each label is characterized by the position of its anchor and the size of its label box. Moreover, the label definition may include a text string and font configuration for each label.

```
{
    "screen": {
        "size": {"width": 2400, "height": 1600},
        "background": {"file": "background.svg"}
    },
    "labels": {
        "definition": {
            "0": {
                "text": "Brno",
                "size": [76, 20],
                "font": {
                    "size": 17,
                    "family": "Arial",
                    "style": "Regular"}
            },
            ...
}
```

[1]The framework Box2D is available at https://box2d.org.

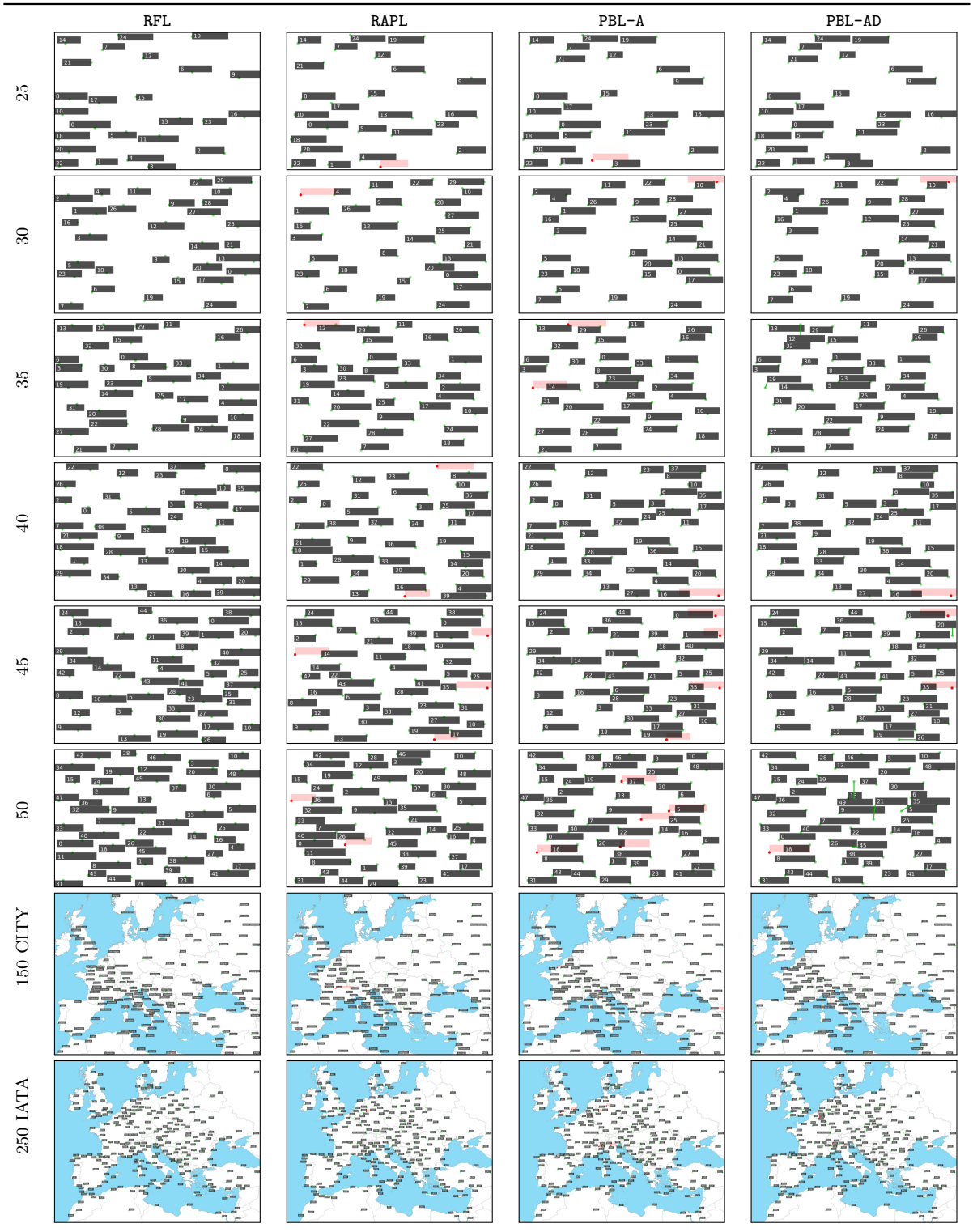


Table B.3: Visual comparison of examined methods applied to selected instances from the compact dataset and real-world instances of IATA airport codes with 250 anchors and CITY names with 150 anchors based on data obtained from Open Street Maps. The green dot represents the anchor (*i.e.*, point feature). The gray rectangle symbolizes the body of the label itself. The red dot describes an anchor that was not labeled by the given method. The red rectangle illustrates the dimensions of the missing label.

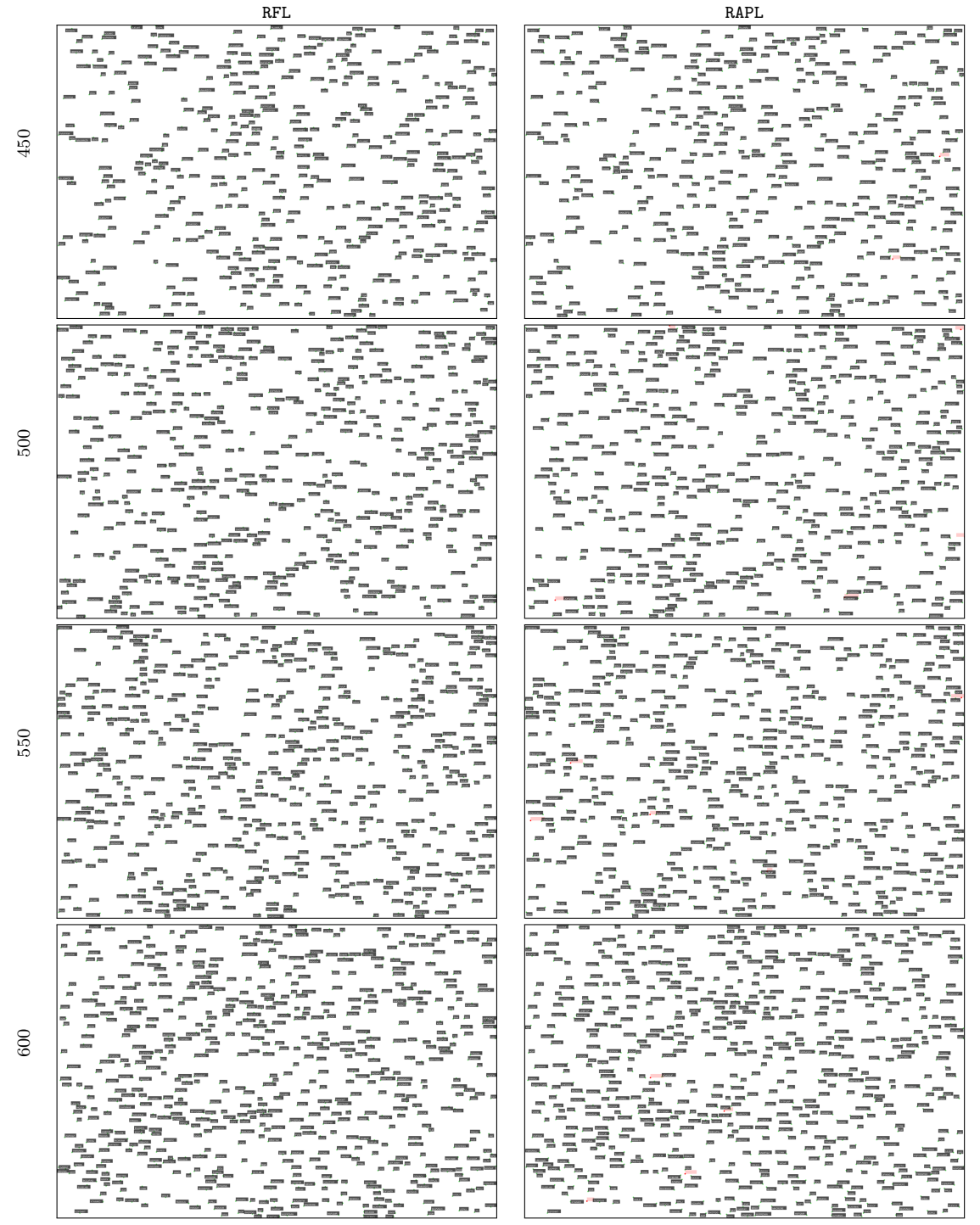


Table B.4: Visual comparison of RFL and RAPL methods applied to selected instances from the volume dataset.

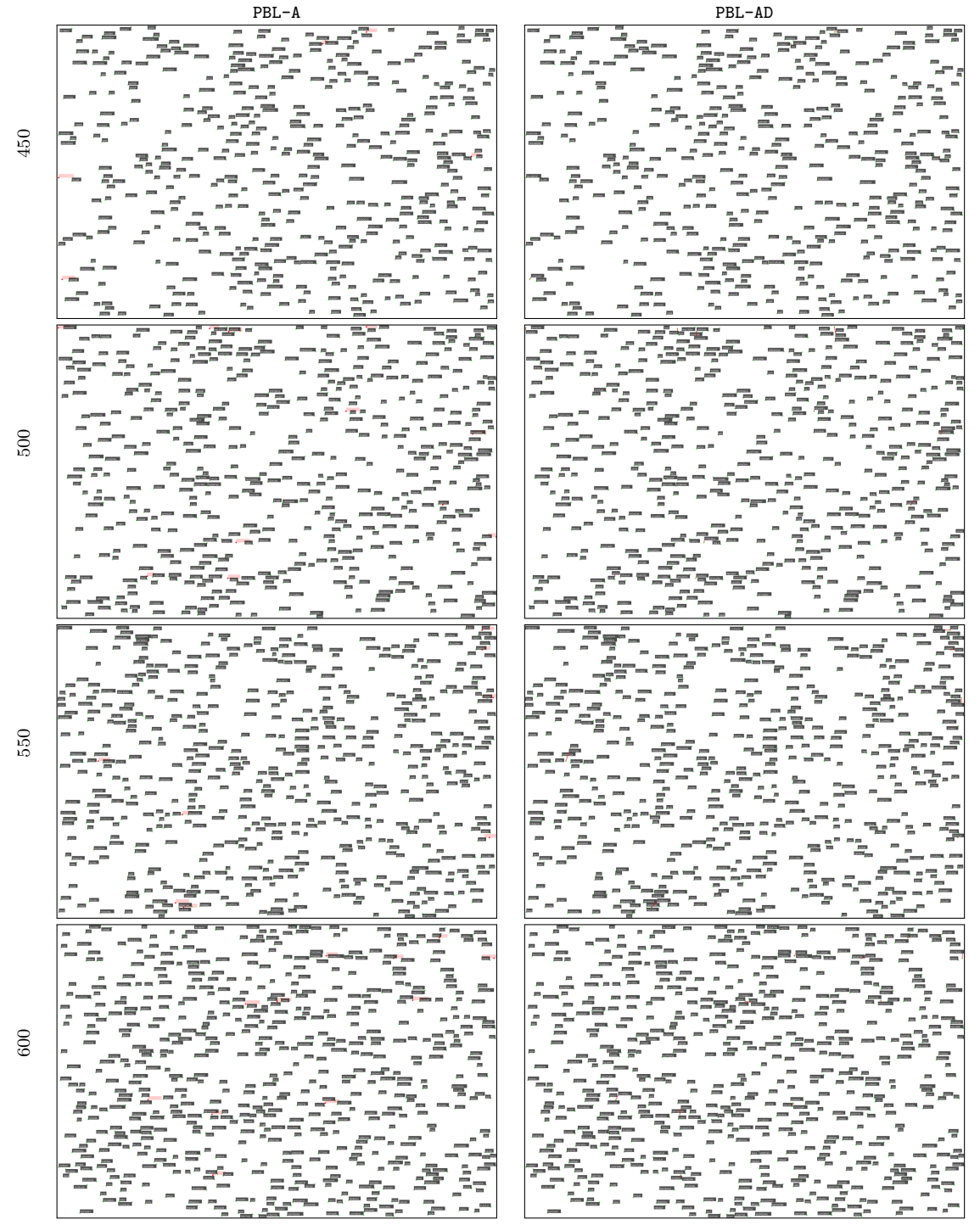
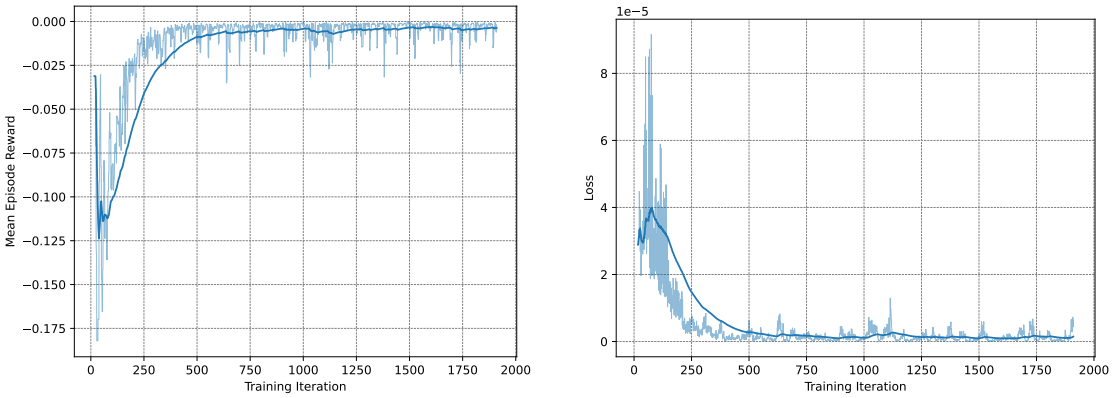


Table B.5: Visual comparison of PBL-A and PBL-AD methods applied to selected instances from the volume dataset.



(a) (b)

Figure B.1: Training metrics of `RFL` (denoted by light blue) smoothed out using Exponential Moving Average (denoted by dark blue). Charts (a) and (b) show *mean episode reward* (same as mean return with discount factor $\gamma = 1.0$) and *loss* over within training iteration.

Appendix C

Perceptual Prioritization of Point-Feature Label Positions

In this appendix, we provide a collection of additional resources, including various images, detailed tables, comprehensive instructions for conducting the experiments, and further results not included in the primary publication. These elements were omitted from the main document in an effort to preserve its conciseness and readability. Our aim is to ensure that readers have access to all necessary supplementary information to fully understand the scope and depth of our research findings.

C.1 `PerceptPPO` Study

Addressing the lack of uniformity in label position prioritization in cartography and GIS, our objective is to establish a PPO that is deeply anchored in user perceptions rather than relying on typographic and cartographic traditions. These traditional practices, though historically significant, stem from methodologies developed decades ago and might not reflect the requirements of today's users. With `PerceptPPO`, our aim is to transcend traditional label placement strategies by leveraging empirical user data to dictate label positioning, thereby enhancing the perceived clarity, usability, and overall user experience of maps.

C.1.1 User Study Interface

This section provides examples of user interface used to establish `PerceptPPO`. The following paragraphs present textual information available to participants as seen in Figure C.1 and Figure C.2.

Introduction. Welcome to our user study! We are excited to have you participate in this research to help us understand more about the perception of maps. Your feedback will play a crucial role in shaping future map-based products and services.

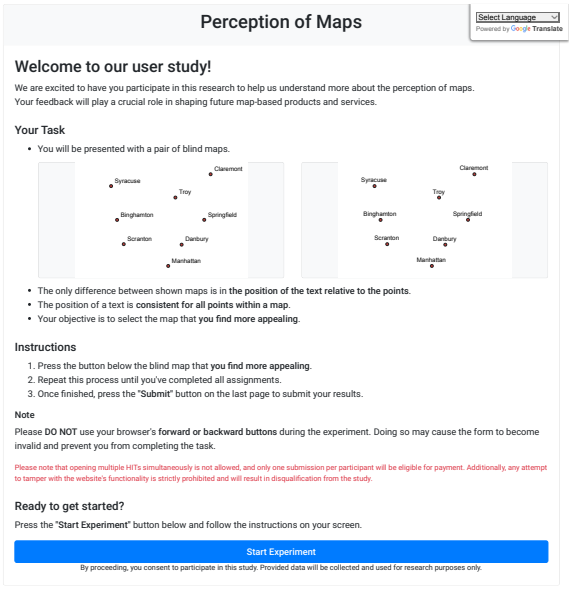
Task. You will be presented with a pair of blind maps. The only difference between shown maps is in the position of the text relative to the points. The position of a text is consistent for all points within a map. Your objective is to select the map that you find more appealing.

Instructions. Press the button below the blind map that you find more appealing. Repeat this process until you've completed all assignments. Once finished, press the „Submit" button on the last page to submit your results.

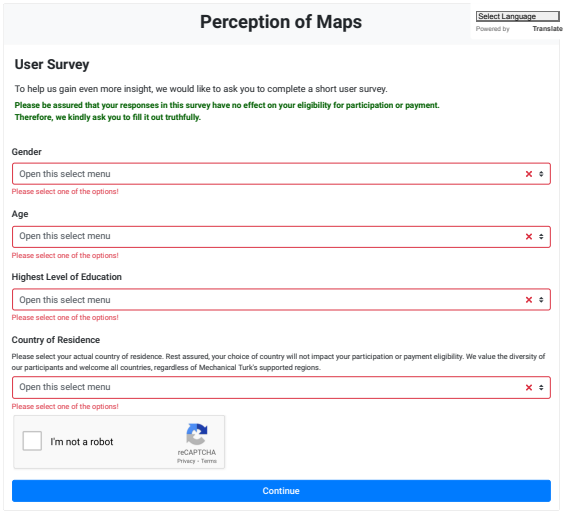
Note. Please note that opening multiple HITs simultaneously is not allowed, and only one submission per participant will be eligible for payment. Additionally, any attempt to tamper with the website's functionality is strictly prohibited and will result in disqualification from the study.

Start Experiment. Ready to get started? Press the „Start Experiment" button below and follow the instructions on your screen. By proceeding, you consent to participate in this study. Provided data will be collected and used for research purposes only.

Call For Action. Which of the two maps do you find more appealing?

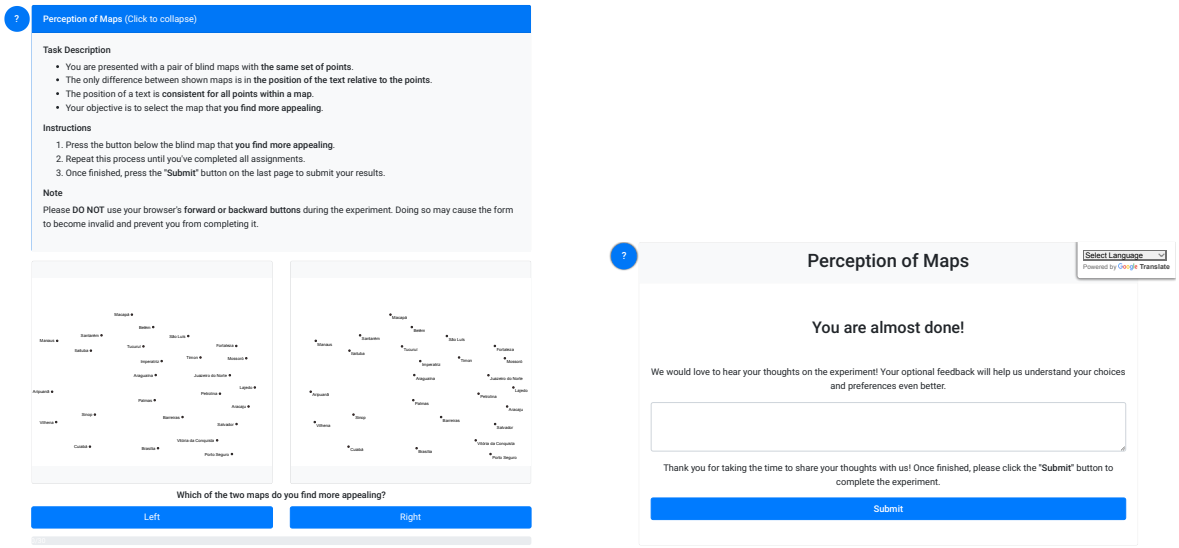


(a) Introduction



(b) Survey

Figure C.1: **User Study – `PerceptPPO`:** Web interface for introduction and survey.



(a) Experiment (b) Feedback

Figure C.2: **User Study – `PerceptPPO`:** Web interface for experiment and feedback.

C.1.2 Data

Figure C.17, Figure C.18, and Figure C.19 display the map renders utilized in the user study, showcasing all examined label positions within the 8-position model. We randomly selected 30 global locations, excluding maritime regions and areas south of -60 degrees latitude to omit Antarctica, given its minimal population. Each site was centered within a zone defined by specific coordinates and a zoom level between 5 (comparable to the size of Europe) and 10 (similar in area to Luxembourg), rendered into vector SVG images measuring 1305×1025 pixels. We identified settlements with populations exceeding 500, using data from GeoNames, specifically the Cities 500 dataset[1], as anchor points. These anchors were ranked by population size, and only those with all eight potential label positions conflict-free were included. Areas with less than 20 suitable anchors were replaced. This process yielded 30 areas, numbered 0 to 29, at zoom levels 5 to 8, featuring between 20 to 54 anchors each.

C.1.3 Results

Table C.1 presents comprehensive p-values from pairwise comparisons of label positions. Additionally, Table C.2, Table C.3, and Table C.4 detail the p-values for the pairwise comparisons within the identified clusters. Table C.9 shows the coefficient of consistency ζ and coefficient of agreement u along with its corresponding values of $u_{\min}$, χ^2 and p-value for positions of `PerceptPPO` for each map area reported over all responses and clusters obtained by hierarchical clustering. Figure C.8 and Figure C.9 show the detailed statistics of engaged participants.

[1] http://download.geonames.org/export/dump/

	T	B	R	TR	BR	L	TL	BL
T	–	1.64e-03	3.81e-07	8.59e-17	3.31e-22	3.49e-35	1.28e-55	4.38e-50
B	1.64e-03	–	6.40e-04	3.22e-08	3.58e-20	1.09e-29	2.93e-34	8.54e-48
R	3.81e-07	6.40e-04	–	1.25e-03	4.46e-12	1.30e-21	2.42e-19	3.39e-31
TR	8.59e-17	3.22e-08	1.25e-03	–	4.57e-04	2.06e-05	6.10e-20	4.82e-24
BR	3.31e-22	3.58e-20	4.46e-12	4.57e-04	–	1.19e-02	1.25e-07	2.43e-20
L	3.49e-35	1.09e-29	1.30e-21	2.06e-05	1.19e-02	–	1.30e-03	2.08e-09
TL	1.28e-55	2.93e-34	2.42e-19	6.10e-20	1.25e-07	1.30e-03	–	5.79e-04
BL	4.38e-50	8.54e-48	3.39e-31	4.82e-24	2.43e-20	2.08e-09	5.79e-04	–

Table C.1: **User Study – `PerceptPPO`:** Detailed overall p values disregard identified clusters for all position pairs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	T	B	R	TR	BR	L	TL	BL
T	–	2.86e-01	3.20e-01	6.10e-25	1.90e-10	9.61e-15	2.66e-48	3.87e-50
B	2.86e-01	–	4.92e-01	5.93e-19	3.71e-14	3.05e-10	3.42e-31	2.54e-61
R	3.20e-01	4.92e-01	–	3.01e-29	1.77e-19	8.66e-13	3.40e-37	1.39e-83
TR	6.10e-25	5.93e-19	3.01e-29	–	2.80e-01	7.63e-02	2.01e-11	1.70e-16
BR	1.90e-10	3.71e-14	1.77e-19	2.80e-01	–	2.66e-01	3.72e-06	1.01e-13
L	9.61e-15	3.05e-10	8.66e-13	7.63e-02	2.66e-01	–	3.06e-11	3.84e-16
TL	2.66e-48	3.42e-31	3.40e-37	2.01e-11	3.72e-06	3.06e-11	–	2.97e-02
BL	3.87e-50	2.54e-61	1.39e-83	1.70e-16	1.01e-13	3.84e-16	2.97e-02	–

Table C.2: **User Study – `PerceptPPO`:** Detailed p values of Cluster 1 for all position pairs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	T	B	R	TR	BR	L	TL	BL
T	–	3.76e-10	1.80e-45	8.22e-14	1.15e-36	3.96e-63	2.57e-31	1.74e-65
B	3.76e-10	–	6.86e-20	5.14e-03	1.85e-18	4.08e-38	1.18e-08	3.09e-27
R	1.80e-45	6.86e-20	–	4.27e-10	7.51e-02	1.12e-05	2.73e-02	4.78e-03
TR	8.22e-14	5.14e-03	4.27e-10	–	1.97e-05	3.44e-13	2.88e-03	1.06e-12
BR	1.15e-36	1.85e-18	7.51e-02	1.97e-05	–	3.26e-06	1.62e-01	8.91e-05
L	3.96e-63	4.08e-38	1.12e-05	3.44e-13	3.26e-06	–	1.73e-14	9.46e-03
TL	2.57e-31	1.18e-08	2.73e-02	2.88e-03	1.62e-01	1.73e-14	–	1.82e-08
BL	1.74e-65	3.09e-27	4.78e-03	1.06e-12	8.91e-05	9.46e-03	1.82e-08	–

Table C.3: **User Study – `PerceptPPO`:** Detailed p values of Cluster 2 for all position pairs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	T	B	R	TR	BR	L	TL	BL
T	–	6.50e-02	2.30e-01	4.07e-02	2.41e-01	9.99e-08	1.71e-10	9.62e-06
B	6.50e-02	–	2.78e-01	1.09e-02	2.29e-01	3.09e-05	6.09e-05	4.62e-04
R	2.30e-01	2.78e-01	–	1.71e-02	4.51e-01	1.16e-04	1.11e-04	4.26e-03
TR	4.07e-02	1.09e-02	1.71e-02	–	3.42e-03	5.87e-06	1.55e-08	1.50e-05
BR	2.41e-01	2.29e-01	4.51e-01	3.42e-03	–	1.40e-04	1.32e-05	9.08e-04
L	9.99e-08	3.09e-05	1.16e-04	5.87e-06	1.40e-04	–	4.59e-01	1.99e-01
TL	1.71e-10	6.09e-05	1.11e-04	1.55e-08	1.32e-05	4.59e-01	–	1.75e-01
BL	9.62e-06	4.62e-04	4.26e-03	1.50e-05	9.08e-04	1.99e-01	1.75e-01	–

Table C.4: **User Study – `PerceptPPO`:** Detailed p values of Cluster 3 for all position pairs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

C.2 Evaluation 1: Label Density

In this evaluation, our objective was to understand the impact of label density on users' preferences. Recognizing that dense labeling can clutter a map and reduce its legibility, while too sparse a distribution might overlook important information, we sought to identify an optimal label density from a user-centered perspective.

C.2.1 User Study Interface

This section provides examples of user interface used to undercover preferred label density. The following paragraphs present textual information available to participants as seen in Figure C.3, Figure C.4, and Figure C.5.

Introduction. Welcome to our user study! We are excited to have you participate in this research to help us understand more about the perception of maps. Your feedback will play a crucial role in shaping future map-based products and services.

Task. You will see a map along with a slider positioned below it. The slider will alter the density (amount) of labels on the map. Your goal is to use the slider to set a label density on the map that feels comfortable and not overwhelming by the amount of information for you. Please do not focus on the specific cities or locations presented, but solely on the density of labels.

Instructions. Move the slider with your mouse or use the left and right arrow keys to set a prefered label density. Press „Continue“ to proceed to the next map. At the end of the assignment, you will have an opportunity to provide additional feedback. Afterward, press the „Submit“ to submit your response.

Note. Please note that opening multiple HITs simultaneously is not allowed, and only one submission per participant will be eligible for payment. Additionally, any attempt to tamper with the website's functionality is strictly prohibited and will result in disqualification from the study.

Start Experiment. Ready to get started? Press the „Start Experiment“ button below and follow the instructions on your screen. By proceeding, you consent to participate in this study. Provided data will be collected and used for research purposes only.

Call For Action. Adjust the slider to choose a label density that you find comfortable without being overwhelmed by the amount of information, disregarding the specific cities displayed.

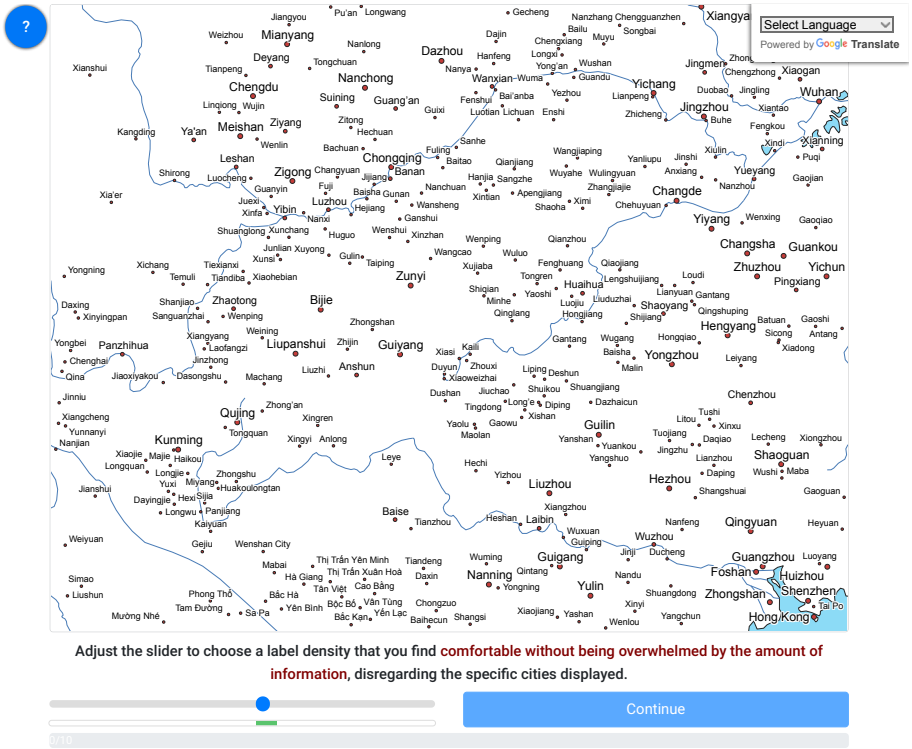


Figure C.3: **Evaluation 1 – Label Density:** Web interface for experiment. Initial position of slider is randomized in each trial.

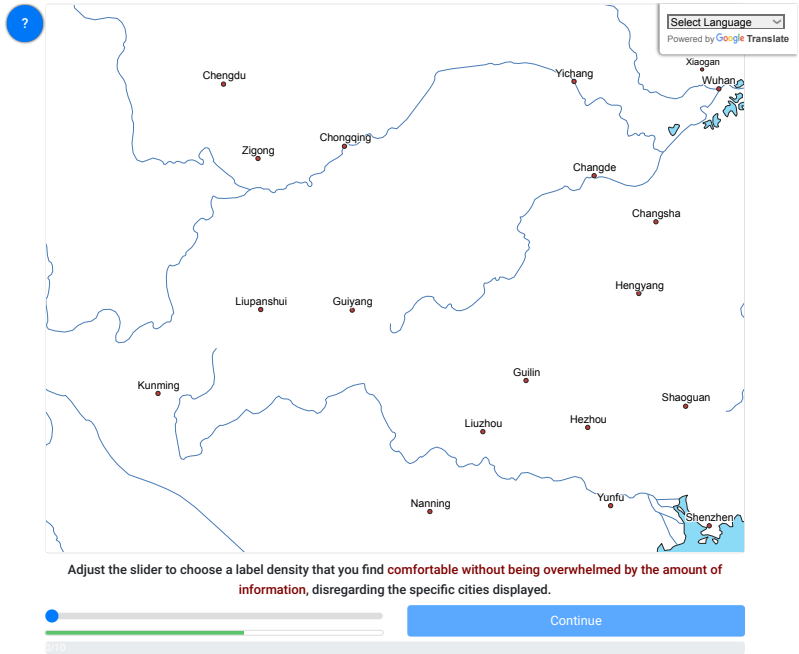


Figure C.4: **Evaluation 1 – Label Density:** Web interface for experiment. After setting the slider to the leftmost position while exploring the intermediate densities, as shown by the green bar below the slider.

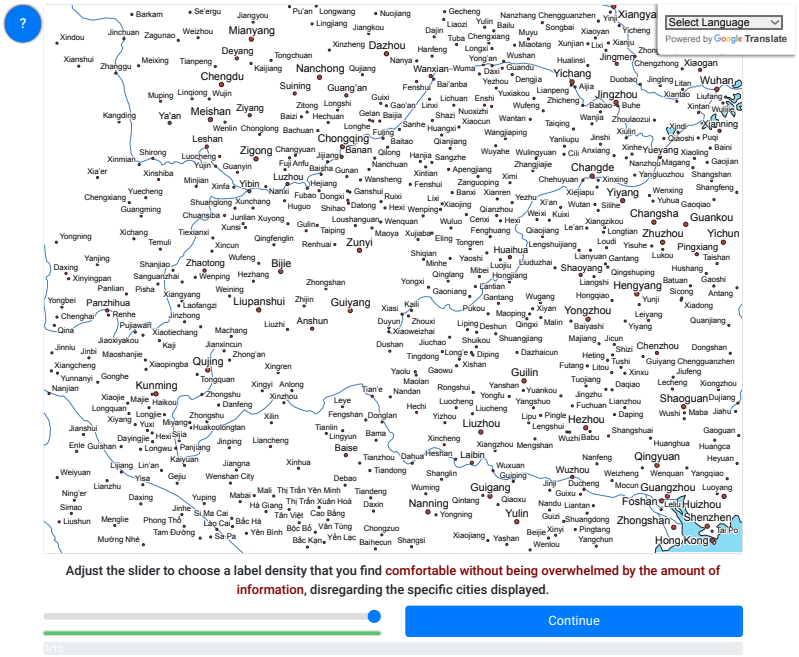


Figure C.5: **Evaluation 1 – Label Density:** Web interface for experiment. After setting the slider to the rightmost position while exploring the intermediate densities, as shown by the green bar below the slider.

C.2.2 Data

Figure C.20, Figure C.21, Figure C.22, Figure C.23, Figure C.24, Figure C.25, Figure C.26, Figure C.27, Figure C.28, and Figure C.29 display the map renders utilized in the label density user study, showcasing various label density thresholds LD_{thr} for all selected areas. We chose ten populated areas (0, 4, 5, 6, 9, 12, 13, 17, 27, 29) from the `PerceptPPO` study to cover a broad spectrum of label density scenarios, including geographical features like continent borders and bodies of water. This approach aimed to mirror maps encountered in everyday life while limiting extraneous influences on participant choices. We utilized Natural Earth data for physical features and maintained the same city data source as in previous sections, opting for a 1:50,000,000 scale to balance detail with manageable SVG image sizes.

C.2.3 Results

Figure C.15 show probabilities that a label is placed at a specific position for each investigated method dependent on label density. Figure C.12 displays frequency of LD_{thr}, $\overline{LLDF}$, and GLD of study for all participants. Figure C.10 and Figure C.11 show the detailed statistics of engaged participants.

C.3 Evaluation 2: Comparison of PPOs

This segment of our research focuses on a comparative analysis of various Position Priority Orders (PPOs), as delineated in our study. We aim to ascertain the relative effectiveness

and user preference across different PPO configurations, offering insights into how label placement strategies can be optimized for user satisfaction.

C.3.1 User Study Interface

This section provides examples of user interface used to compare PPOs. The following paragraphs present textual information available to participants as seen in Figure C.6 and Figure C.7.

Introduction. Welcome to our user study! We are excited to have you participate in this research to help us understand more about the perception of maps. Your feedback will play a crucial role in shaping future map-based products and services.

Task. You will be presented with a pair of blind maps. The difference between shown maps is in the position of the text relative to the points. Your goal is to select the map that you like more. Please ignore specific cities or locations presented, as the maps are not required to display the same set of points. If both maps look identical and you can not decide, please choose one randomly.

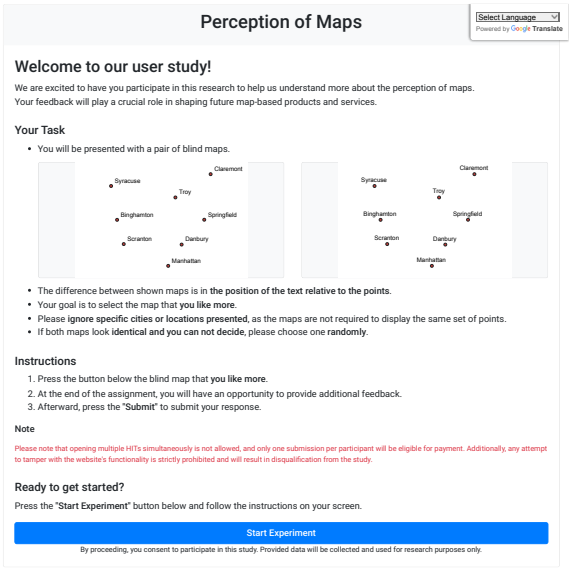
Instructions. Press the button below the blind map that you like more. At the end of the assignment, you will have an opportunity to provide additional feedback. Afterward, press the „Submit“ to submit your response.

Note. Please note that opening multiple HITs simultaneously is not allowed, and only one submission per participant will be eligible for payment. Additionally, any attempt to tamper with the website's functionality is strictly prohibited and will result in disqualification from the study.

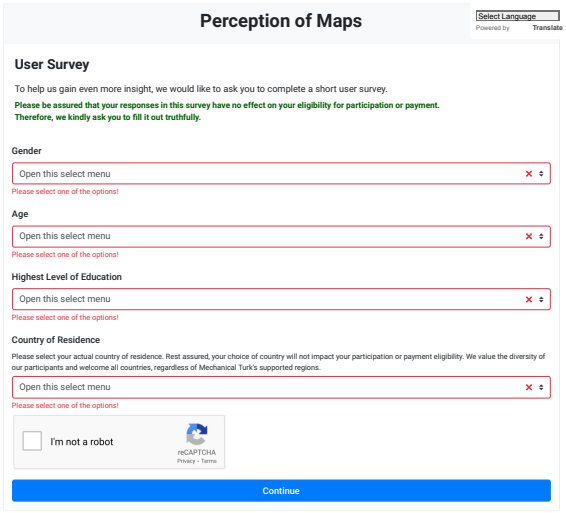
Start Experiment. Ready to get started? Press the „Start Experiment“ button below and follow the instructions on your screen. By proceeding, you consent to participate in this study. Provided data will be collected and used for research purposes only.

Survey. To help us gain even more insight, we would like to ask you to complete a short user survey. Please be assured that your responses in this survey have no effect on your eligibility for participation or payment. Therefore, we kindly ask you to fill it out truthfully.

Call For Action. Which of the two maps do you prefer? Focus mainly on the position of the text relative to the points.

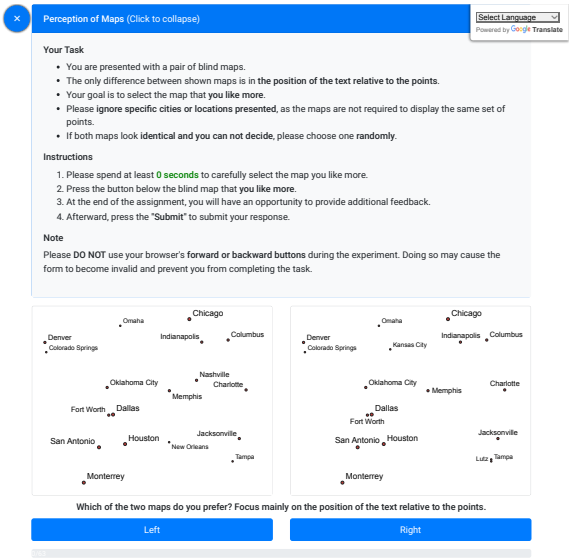


(a) Introduction

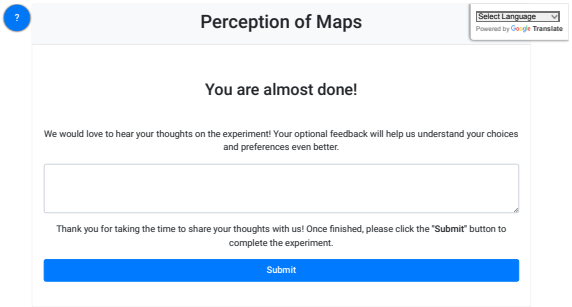


(b) Survey

Figure C.6: **Evaluation 2 – Comparison Study of PPOs:** Web interface for introduction and survey.



(a) Experiment



(b) Feedback

Figure C.7: **Evaluation 2 – Comparison Study of PPOs:** Web interface for experiment and survey.

C.3.2 Data

Figure C.30, Figure C.31, and Figure C.32 display the map renders utilized in the PPO comparison user study, showcasing all examined PPOs.

C.3.3 Results

Table C.5 presents comprehensive p-values from pairwise comparisons of examined PPOs. Additionally, Table C.6, Table C.7, and Table C.8 detail the p-values for the pairwise comparisons within the identified clusters. Table C.10 shows coefficient of consistency ζ and coefficient of agreement u along with its corresponding values of $u_{\min}$, χ^2 and p-value for examined PPOs for each map area reported over all responses and clusters obtained by hierarchical clustering. Figure C.13 and Figure C.14 show the detailed statistics of engaged participants. Figure C.16 show probabilities that a label is placed at a specific position for each investigated method for the data used in the PPO comparison user study (652×512 size of renders and label density $LD_{\text{thr}} = 12.5\%$).

	Brewer	YoeliB	Christensen	Slocum	Imhof	Zoraster	PerceptPPO
Brewer	–	2.37e-01	1.80e-02	4.45e-01	6.72e-02	4.68e-11	8.35e-14
YoeliB	2.37e-01	–	8.60e-02	2.34e-01	2.42e-02	1.35e-11	2.09e-15
Christensen	1.80e-02	8.60e-02	–	1.85e-02	3.38e-04	7.71e-14	7.07e-18
Slocum	4.45e-01	2.34e-01	1.85e-02	–	9.62e-02	1.79e-10	3.41e-13
Imhof	6.72e-02	2.42e-02	3.38e-04	9.62e-02	–	5.29e-10	6.71e-13
Zoraster	4.68e-11	1.35e-11	7.71e-14	1.79e-10	5.29e-10	–	3.35e-02
PerceptPPO	8.35e-14	2.09e-15	7.07e-18	3.41e-13	6.71e-13	3.35e-02	–

Table C.5: **Evaluation 2 – Comparison Study of PPOs:** Detailed overall p values disregard identified clusters for the examined PPOs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	Brewer	YoeliB	Christensen	Slocum	Imhof	Zoraster	PerceptPPO
Brewer	–	1.08e-03	2.62e-06	8.13e-03	1.43e-03	1.97e-57	1.73e-49
YoeliB	1.08e-03	–	8.01e-02	2.08e-01	3.75e-07	7.23e-98	1.09e-80
Christensen	2.62e-06	8.01e-02	–	1.51e-02	6.79e-12	2.42e-105	9.26e-88
Slocum	8.13e-03	2.08e-01	1.51e-02	–	4.39e-06	1.26e-72	1.43e-66
Imhof	1.43e-03	3.75e-07	6.79e-12	4.39e-06	–	1.30e-50	1.07e-41
Zoraster	1.97e-57	7.23e-98	2.42e-105	1.26e-72	1.30e-50	–	4.24e-01
PerceptPPO	1.73e-49	1.09e-80	9.26e-88	1.43e-66	1.07e-41	4.24e-01	–

Table C.6: **Evaluation 2 – Comparison Study of PPOs:** Detailed p values of Cluster 1 for the examined PPOs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	Brewer	YoeliB	Christensen	Slocum	Imhof	Zoraster	PerceptPPO
Brewer	–	5.45e-03	9.59e-03	1.71e-01	1.12e-01	8.18e-38	3.94e-29
YoeliB	5.45e-03	–	4.76e-01	5.57e-04	2.77e-01	9.61e-22	3.83e-21
Christensen	9.59e-03	4.76e-01	–	1.47e-03	3.08e-01	2.08e-26	3.26e-24
Slocum	1.71e-01	5.57e-04	1.47e-03	–	2.41e-02	7.05e-39	2.14e-33
Imhof	1.12e-01	2.77e-01	3.08e-01	2.41e-02	–	1.59e-27	2.30e-33
Zoraster	8.18e-38	9.61e-22	2.08e-26	7.05e-39	1.59e-27	–	1.56e-01
PerceptPPO	3.94e-29	3.83e-21	3.26e-24	2.14e-33	2.30e-33	1.56e-01	–

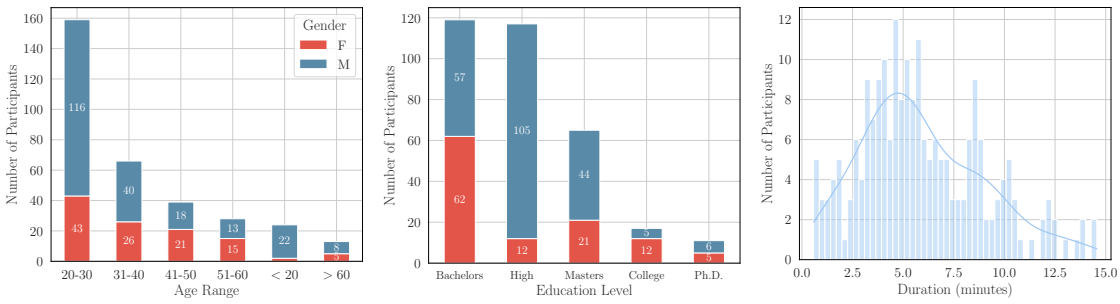
Table C.7: **Evaluation 2 – Comparison Study of PPOs:** Detailed p values of Cluster 2 for the examined PPOs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

	Brewer	YoeliB	Christensen	Slocum	Imhof	Zoraster	PerceptPPO
Brewer	–	4.52e-04	2.58e-02	5.20e-02	4.72e-01	2.77e-03	3.58e-08
YoeliB	4.52e-04	–	1.65e-01	1.63e-01	3.80e-04	4.55e-01	2.92e-03
Christensen	2.58e-02	1.65e-01	–	4.56e-01	1.16e-02	1.36e-01	5.91e-05
Slocum	5.20e-02	1.63e-01	4.56e-01	–	3.25e-02	1.10e-01	5.80e-05
Imhof	4.72e-01	3.80e-04	1.16e-02	3.25e-02	–	2.41e-04	5.84e-13
Zoraster	2.77e-03	4.55e-01	1.36e-01	1.10e-01	2.41e-04	–	4.53e-03
PerceptPPO	3.58e-08	2.92e-03	5.91e-05	5.80e-05	5.84e-13	4.53e-03	–

Table C.8: **Evaluation 2 – Comparison Study of PPOs:** p values of Cluster 3 for the examined PPOs. Green cells indicate statistically significant differences, or red otherwise. We use a Two-Tailed test with $\alpha = 0.05$; therefore, p has to be less than 0.025 to be significant.

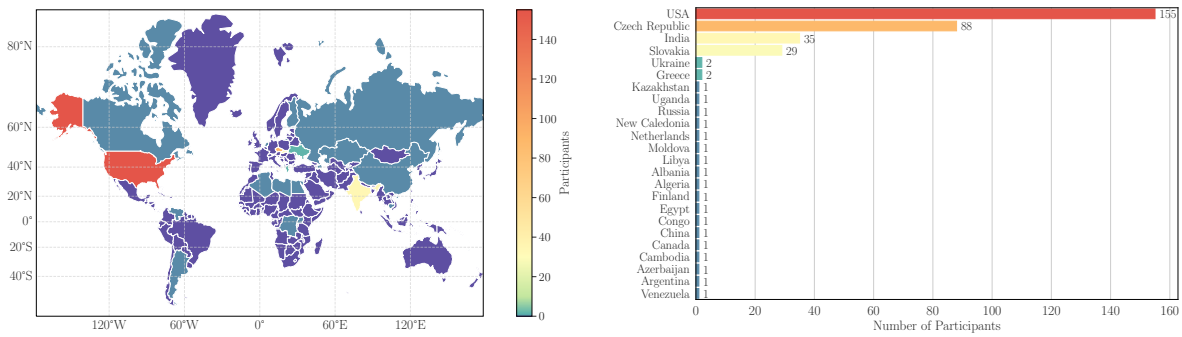
Area	Overall					Cluster 1					Cluster 2					Cluster 3				
	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value
0	0.720	0.142	-0.043	127.937	2.2e-13	0.900	0.310	-0.091	150.960	1.0e-15	0.750	0.286	-0.333	212.000	0.012	0.438	0.013	-0.143	46.222	0.363
1	0.619	0.102	-0.059	87.969	8.7e-07	0.821	0.293	-0.143	115.040	1.3e-07	0.767	0.381	-0.333	228.000	0.001	0.388	-0.018	-0.143	38.222	0.700
2	0.679	0.152	-0.048	126.681	4.9e-13	0.754	0.316	-0.091	153.360	4.0e-16	0.812	0.250	-0.333	124.000	0.003	0.390	-0.071	-0.200	47.556	0.915
3	0.652	0.102	-0.040	106.019	6.3e-10	0.739	0.210	-0.111	101.143	5.8e-07	0.879	0.367	-0.143	132.640	4.4e-10	0.389	0.008	-0.111	42.857	0.398
4	0.625	0.102	-0.048	98.340	1.4e-08	0.819	0.311	-0.143	124.222	1.1e-09	0.733	0.095	-0.333	180.000	0.250	0.455	0.070	-0.091	61.580	0.009
5	0.474	0.044	-0.040	63.932	6.1e-04	0.731	0.186	-0.143	91.556	2.9e-05	0.867	0.286	-0.333	212.000	0.012	0.243	0.010	-0.077	39.389	0.296
6	0.688	0.099	-0.048	93.839	7.6e-08	0.894	0.480	-0.143	168.222	1.5e-16	0.925	0.488	-0.333	164.000	4.2e-07	0.400	0.012	-0.111	44.000	0.352
7	0.700	0.148	-0.048	124.155	1.3e-12	0.861	0.448	-0.111	169.714	1.5e-17	0.975	0.571	-1.000	inf	0	0.500	0.019	-0.111	44.875	0.252
8	0.774	0.116	-0.037	122.170	1.2e-12	0.896	0.319	-0.077	162.281	6.3e-18	0.810	0.371	-0.200	130.222	1.0e-06	0.578	-0.028	-0.111	32.571	0.828
9	0.647	0.127	-0.059	98.116	3.6e-08	0.825	0.343	-0.200	123.500	1.2e-07	0.850	0.143	-0.333	188.000	0.139	0.438	0.074	-0.143	62.222	0.033
10	0.594	0.094	-0.059	83.719	3.4e-06	0.780	0.257	-0.200	108.889	2.3e-04	0.850	0.250	-0.333	124.000	0.003	0.378	0.044	-0.111	53.143	0.099
11	0.664	0.114	-0.048	105.940	9.4e-10	0.744	0.226	-0.111	105.714	1.4e-07	0.825	0.352	-0.200	125.500	6.5e-08	0.421	0.034	-0.143	54.240	0.219
12	0.615	0.167	-0.043	144.698	3.1e-16	0.722	0.317	-0.111	132.000	1.9e-11	0.790	0.457	-0.200	146.222	1.0e-08	0.411	0.111	-0.111	72.571	0.002
13	0.714	0.136	-0.048	119.940	5.4e-12	0.905	0.431	-0.091	185.136	3.1e-21	0.933	0.333	-0.333	220.000	0.004	0.369	-0.010	-0.143	40.222	0.616
14	0.640	0.165	-0.034	170.299	4.3e-21	0.800	0.357	-0.077	177.554	1.4e-20	0.900	0.524	-0.333	252.000	2.9e-05	0.419	0.046	-0.077	53.917	0.029
15	0.717	0.127	-0.043	117.651	1.1e-11	0.911	0.480	-0.111	178.857	4.3e-19	0.770	0.400	-0.200	135.556	2.3e-07	0.494	0.032	-0.111	49.714	0.169
16	0.661	0.114	-0.043	108.889	2.8e-10	0.764	0.361	-0.091	161.136	4.3e-17	0.900	0.429	-0.333	236.000	4.2e-04	0.456	0.012	-0.111	44.000	0.352
17	0.674	0.093	-0.059	81.049	9.6e-06	0.925	0.438	-0.200	143.500	2.2e-10	0.800	0.429	-0.333	236.000	4.2e-04	0.438	-0.028	-0.143	35.556	0.801
18	0.773	0.179	-0.040	166.931	2.5e-20	0.850	0.286	-0.111	128.875	1.9e-11	0.925	0.548	-0.143	186.222	1.5e-19	0.525	0.122	-0.143	74.889	0.002
19	0.673	0.105	-0.040	111.264	8.3e-11	0.810	0.295	-0.111	131.875	6.6e-12	0.880	0.314	-0.200	119.556	1.7e-05	0.455	0.016	-0.091	42.914	0.270
20	0.656	0.094	-0.040	100.454	4.7e-09	0.855	0.444	-0.111	178.875	1.1e-19	0.670	0.257	-0.200	108.889	2.3e-04	0.450	0.041	-0.111	51.875	0.088
21	0.725	0.134	-0.059	104.969	2.6e-09	0.939	0.294	-0.111	125.143	2.1e-10	0.767	0.429	-0.333	236.000	4.2e-04	0.383	-0.043	-0.200	42.500	0.836
22	0.636	0.040	-0.048	57.740	0.004	0.867	0.224	-0.200	98.500	1.3e-04	0.925	0.179	-0.333	112.000	0.022	0.425	-0.015	-0.091	30.960	0.746
23	0.758	0.168	-0.043	149.570	3.8e-17	0.910	0.256	-0.111	119.375	5.5e-10	0.808	0.405	-0.200	136.500	2.1e-09	0.531	0.122	-0.143	74.889	0.002
24	0.681	0.076	-0.048	79.102	9.6e-06	0.875	0.390	-0.143	144.889	8.2e-13	0.875	0.357	-1.000	inf	0	0.505	-0.013	-0.091	33.136	0.695
25	0.626	0.125	-0.043	116.127	1.9e-11	0.840	0.310	-0.111	136.375	1.3e-12	0.900	0.143	-1.000	inf	0	0.382	0.026	-0.091	46.469	0.164
26	0.715	0.136	-0.043	123.365	1.3e-12	0.791	0.286	-0.091	135.358	7.7e-13	0.863	0.476	-0.333	162.000	6.9e-07	0.537	0.112	-0.143	72.222	0.004
27	0.722	0.150	-0.053	121.506	4.2e-12	0.837	0.339	-0.143	131.556	9.1e-11	0.833	0.348	-0.200	124.500	8.8e-08	0.458	0.105	-0.200	73.500	0.029
28	0.740	0.130	-0.040	126.193	3.3e-13	0.875	0.281	-0.111	127.375	3.3e-11	0.850	0.305	-0.200	115.500	1.2e-06	0.517	0.147	-0.111	82.857	1.3e-04
29	0.636	0.132	-0.040	128.280	1.5e-13	0.767	0.292	-0.091	144.560	1.2e-14	0.983	0.714	-0.333	284.000	5.6e-08	0.375	0.003	-0.111	39.875	0.448

Table C.9: **`PerceptPPO` Study:** Coefficient of consistency ζ and coefficient of agreement u along with its corresponding values of $u_{\min}$, χ^2 and p-value for positions of `PerceptPPO` for each map area reported over all responses and clusters obtained by hierarchical clustering.



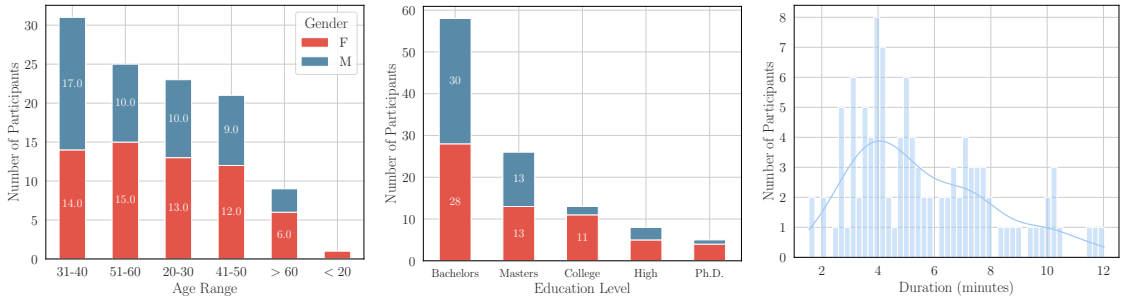
(a) (b) (c)

Figure C.8: **`PerceptPPO` Study:** User statistics of age, education, and and duration of study for all participants.



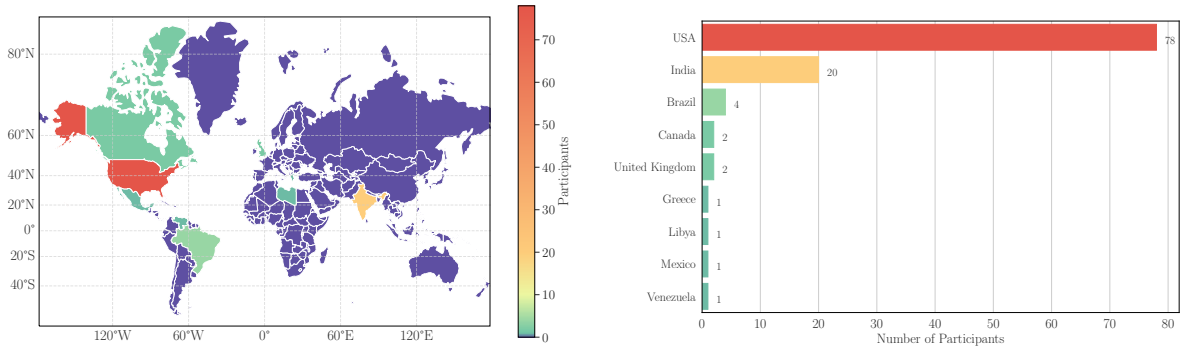
(a) (b)

Figure C.9: **`PerceptPPO` Study:** Country distribution for all participants.



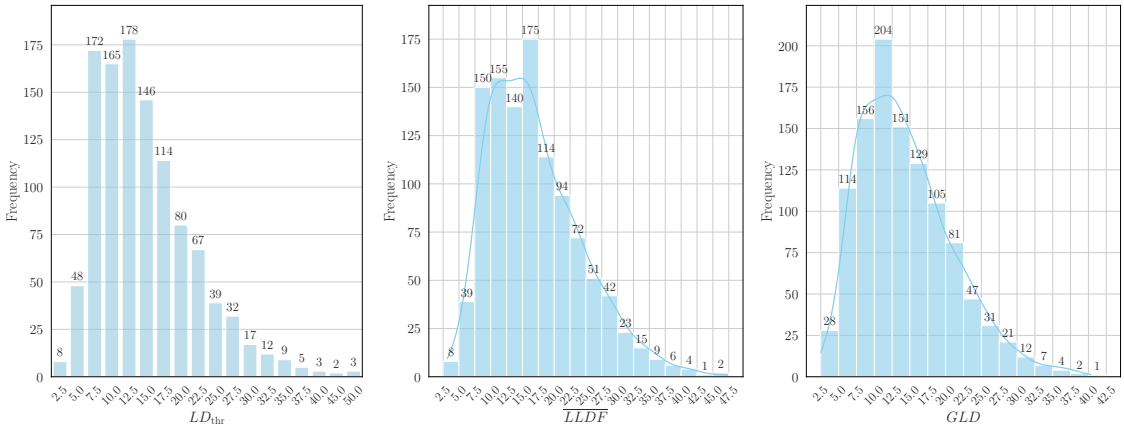
(a) (b) (c)

Figure C.10: **Evaluation 1 – Label Density Study:** User statistics of age, education, and duration of study for all participants.



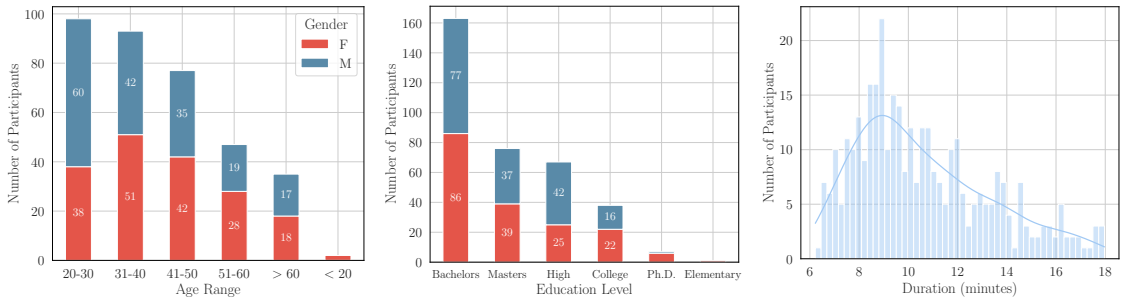
(a) (b)

Figure C.11: **Evaluation 1 – Label Density Study:** Country distribution for all participants.



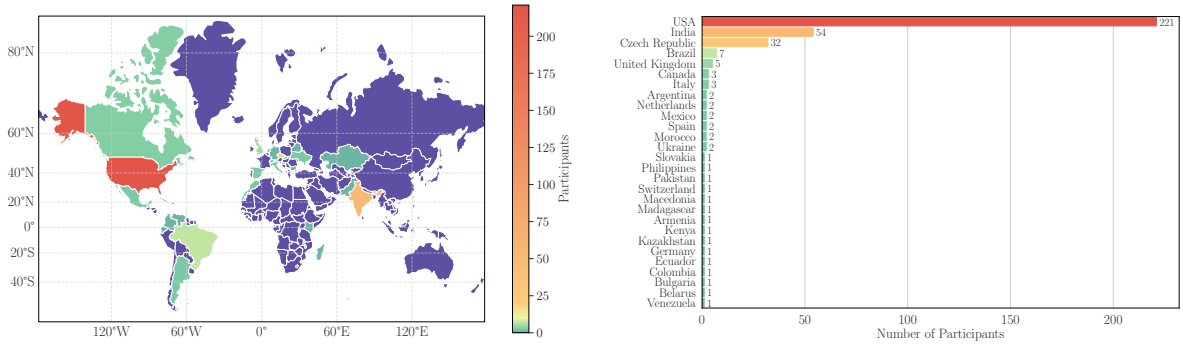
(a) (b) (c)

Figure C.12: **Evaluation 1 – Label Density Study:** Frequency of LD_{thr}, $\overline{LLDF}$, and GLD of study for all participants.



(a) (b) (c)

Figure C.13: **Evaluation 2 – Comparison Study of PPOs:** User statistics of age, education, and duration of study for all participants.



(a) (b)

Figure C.14: **Evaluation 2 – Comparison Study of PPOs:** Country distribution for all participants.

Area	Overall					Cluster 1					Cluster 2					Cluster 3				
	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value	mean ζ	u	$u_{\min}$	χ^2	p-value
0	0.639	0.001	-0.030	23.947	0.407	0.776	0.167	-0.067	82.556	8.9e-08	0.735	0.356	-0.143	97.280	1.0e-07	0.411	0.032	-0.091	36.120	0.132
1	0.599	0.055	-0.029	65.478	5.6e-06	0.684	0.263	-0.053	135.557	3.2e-17	0.536	0.067	-0.200	48.875	0.143	0.487	0.075	-0.091	47.407	0.015
2	0.550	0.038	-0.030	50.572	7.8e-04	0.652	0.253	-0.067	111.479	1.6e-12	0.633	0.048	-0.143	42.880	0.177	0.375	-0.007	-0.091	25.320	0.596
3	0.608	0.002	-0.027	23.977	0.396	0.675	0.243	-0.111	82.857	1.2e-06	0.643	0.165	-0.091	70.519	2.0e-05	0.550	0.013	-0.059	29.920	0.244
4	0.560	-0.006	-0.029	18.419	0.729	0.595	0.134	-0.091	64.520	9.2e-05	0.587	0.079	-0.111	47.429	0.029	0.514	-0.021	-0.067	18.556	0.858
5	0.567	0.010	-0.029	30.523	0.133	0.638	0.233	-0.067	104.710	2.2e-11	0.541	0.084	-0.143	49.280	0.059	0.500	-0.028	-0.077	18.347	0.895
6	0.599	0.020	-0.029	38.183	0.023	0.611	0.114	-0.059	70.727	3.2e-06	0.643	0.048	-0.143	41.333	0.142	0.543	0.024	-0.111	34.781	0.232
7	0.582	0.045	-0.030	54.947	2.1e-04	0.670	0.199	-0.077	85.983	4.8e-08	0.690	0.048	-0.200	45.875	0.221	0.454	0.007	-0.077	28.375	0.368
8	0.601	0.041	-0.029	54.536	2.1e-04	0.714	0.298	-0.077	121.042	5.0e-14	0.667	0.384	-0.200	98.875	5.0e-07	0.478	0.029	-0.067	36.000	0.086
9	0.573	0.018	-0.029	36.099	0.040	0.607	0.111	-0.111	55.281	0.003	0.686	0.314	-0.200	89.333	1.7e-04	0.529	0.024	-0.053	35.074	0.079
10	0.589	0.039	-0.032	49.947	1.0e-03	0.729	0.306	-0.111	101.281	8.6e-10	0.540	0.069	-0.111	45.143	0.047	0.516	0.060	-0.077	44.529	0.019
11	0.547	0.029	-0.029	44.826	0.004	0.656	0.110	-0.091	56.296	0.001	0.589	0.224	-0.143	76.000	2.6e-05	0.451	0.025	-0.067	34.571	0.114
12	0.532	0.015	-0.029	34.183	0.061	0.669	0.193	-0.091	77.630	2.0e-06	0.500	0.072	-0.077	48.165	0.008	0.440	-0.007	-0.091	25.320	0.596
13	0.675	0.072	-0.029	79.007	4.3e-08	0.795	0.356	-0.067	146.556	9.7e-19	0.679	0.276	-0.200	81.875	8.2e-05	0.552	0.015	-0.067	30.864	0.237
14	0.589	0.030	-0.029	45.948	0.003	0.590	0.164	-0.067	81.325	1.4e-07	0.673	0.156	-0.143	62.080	0.004	0.546	-0.002	-0.077	25.708	0.509
15	0.577	0.017	-0.029	35.830	0.042	0.675	0.244	-0.059	122.727	7.9e-15	0.520	0.111	-0.143	54.080	0.022	0.455	-0.004	-0.091	26.963	0.548
16	0.604	0.046	-0.027	59.634	4.0e-05	0.729	0.320	-0.067	134.249	1.6e-16	0.774	0.359	-0.200	94.875	1.8e-06	0.424	-0.006	-0.067	23.429	0.593
17	0.608	0.006	-0.029	27.614	0.228	0.665	0.326	-0.077	123.802	2.5e-14	0.550	0.007	-0.111	30.781	0.402	0.595	-0.016	-0.091	22.920	0.724
18	0.504	0.005	-0.029	26.301	0.281	0.533	0.092	-0.077	53.983	0.002	0.476	0.003	-0.200	38.875	0.493	0.492	0.048	-0.059	43.253	0.015
19	0.623	0.041	-0.029	55.007	1.8e-04	0.743	0.282	-0.067	121.325	3.1e-14	0.531	0.011	-0.143	36.480	0.413	0.541	0.014	-0.077	30.708	0.263
20	0.591	0	-0.029	22.772	0.468	0.770	0.285	-0.077	117.042	2.5e-13	0.487	0.120	-0.091	58.963	6.8e-04	0.468	0.027	-0.091	34.963	0.189
21	0.552	-0.006	-0.027	18.034	0.747	0.539	0.238	-0.091	89.185	3.6e-08	0.667	0.240	-0.091	93.720	4.3e-09	0.464	0.015	-0.077	31.042	0.249
22	0.553	0.003	-0.029	25.433	0.326	0.634	0.224	-0.143	76.000	2.6e-05	0.684	0.011	-0.143	36.480	0.413	0.475	0.011	-0.053	29.296	0.236
23	0.580	0.013	-0.029	32.463	0.090	0.625	0.245	-0.091	95.320	2.4e-09	0.643	0.143	-0.111	62.781	3.4e-04	0.489	-0.018	-0.077	21.256	0.777
24	0.584	0.044	-0.029	56.463	1.2e-04	0.589	0.196	-0.067	96.000	4.8e-10	0.464	0.143	-0.200	60.875	0.016	0.632	0.065	-0.077	45.983	0.013
25	0.635	0.011	-0.029	31.242	0.114	0.612	0.161	-0.067	83.429	5.1e-08	0.736	0.081	-0.111	48.281	0.016	0.571	-0.007	-0.111	27.281	0.584
26	0.557	8.8e-04	-0.029	23.493	0.429	0.571	0.233	-0.077	96.165	1.1e-09	0.565	0.201	-0.091	82.920	2.1e-07	0.529	0.029	-0.111	35.781	0.198
27	0.627	-4.0e-04	-0.029	22.523	0.486	0.708	0.352	-0.091	118.519	6.1e-13	0.706	0.201	-0.111	73.714	2.3e-05	0.519	4.5e-04	-0.067	25.941	0.472
28	0.612	0.018	-0.032	35.680	0.046	0.643	0.231	-0.091	87.407	6.8e-08	0.619	0.212	-0.111	76.000	1.1e-05	0.577	0.053	-0.091	42.120	0.039
29	0.605	0.018	-0.029	36.889	0.032	0.633	0.214	-0.077	94.375	1.6e-09	0.698	0.132	-0.111	58.857	0.002	0.511	0.009	-0.077	29.256	0.352

Table C.10: **Evaluation 2 – Comparison Study of PPOs:** Coefficient of consistency ζ and coefficient of agreement u along with its corresponding values of $u_{\min}$, χ^2 and p-value for examined PPOs for each map area reported over all responses and clusters obtained by hierarchical clustering.

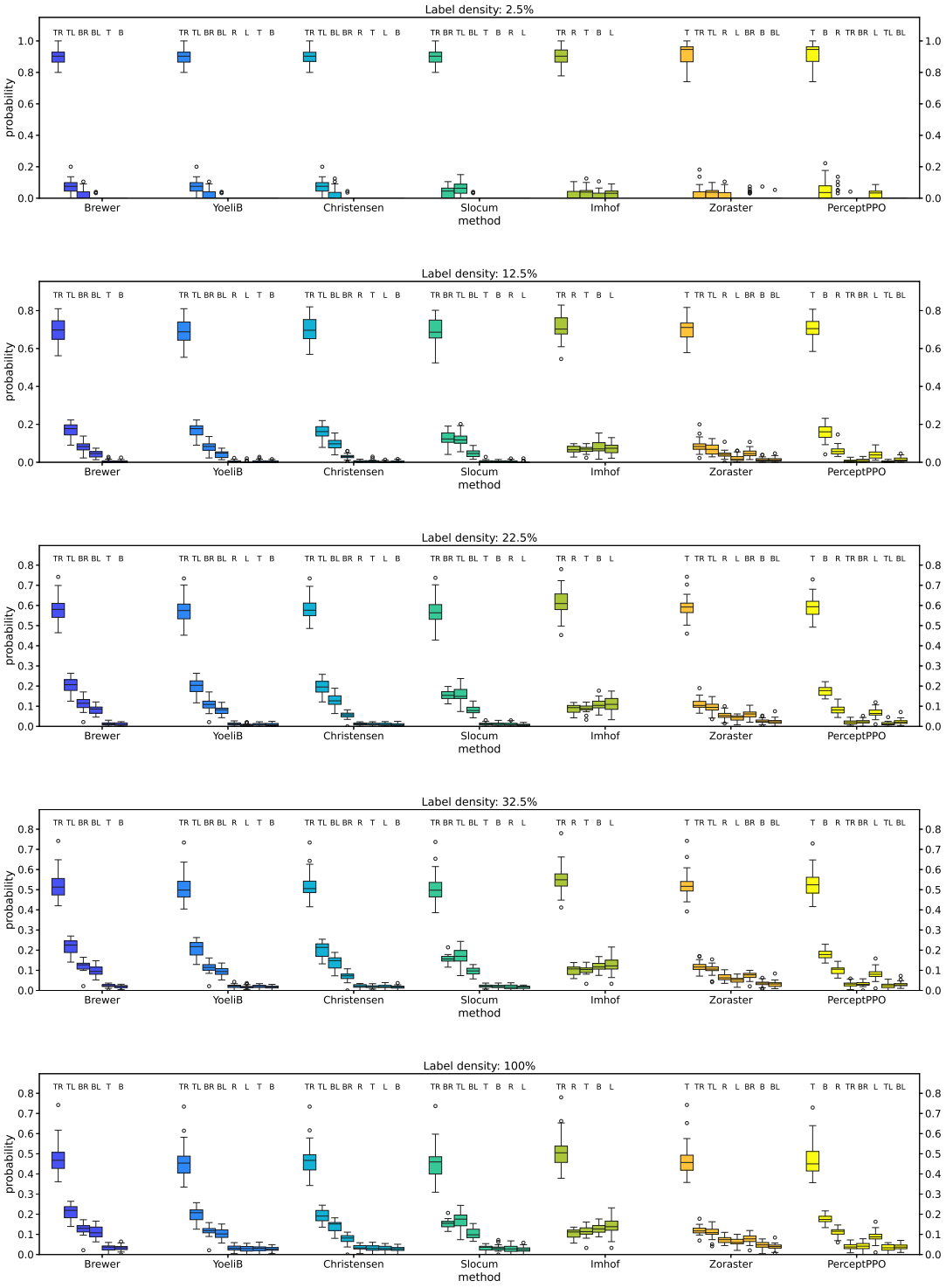


Figure C.15: **Evaluation 1 – Label Density Study:** The probability that a label is placed at a specific position for each investigated method is dependent on label density. For each PPO, the positions (see the labels above the boxplots) are in ascending order with respect to their priorities from left to right. Label density threshold LD_{thr} increases from top to bottom.

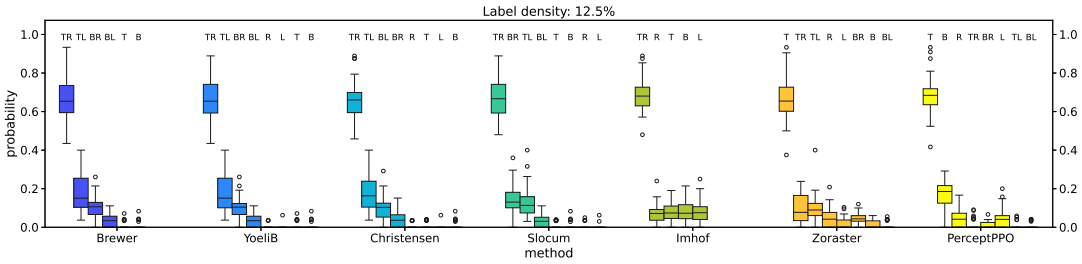


Figure C.16: **Evaluation 2 – Comparison Study of PPOs:** The probability that a label is placed at a specific position for each investigated method for the data used in Evaluation 2 (652×512 size of renders and label density $LD_{\text{thr}} = 12.5\%$). For each PPO, the positions (see the labels above the boxplots) are in ascending order with respect to their priorities from left to right.

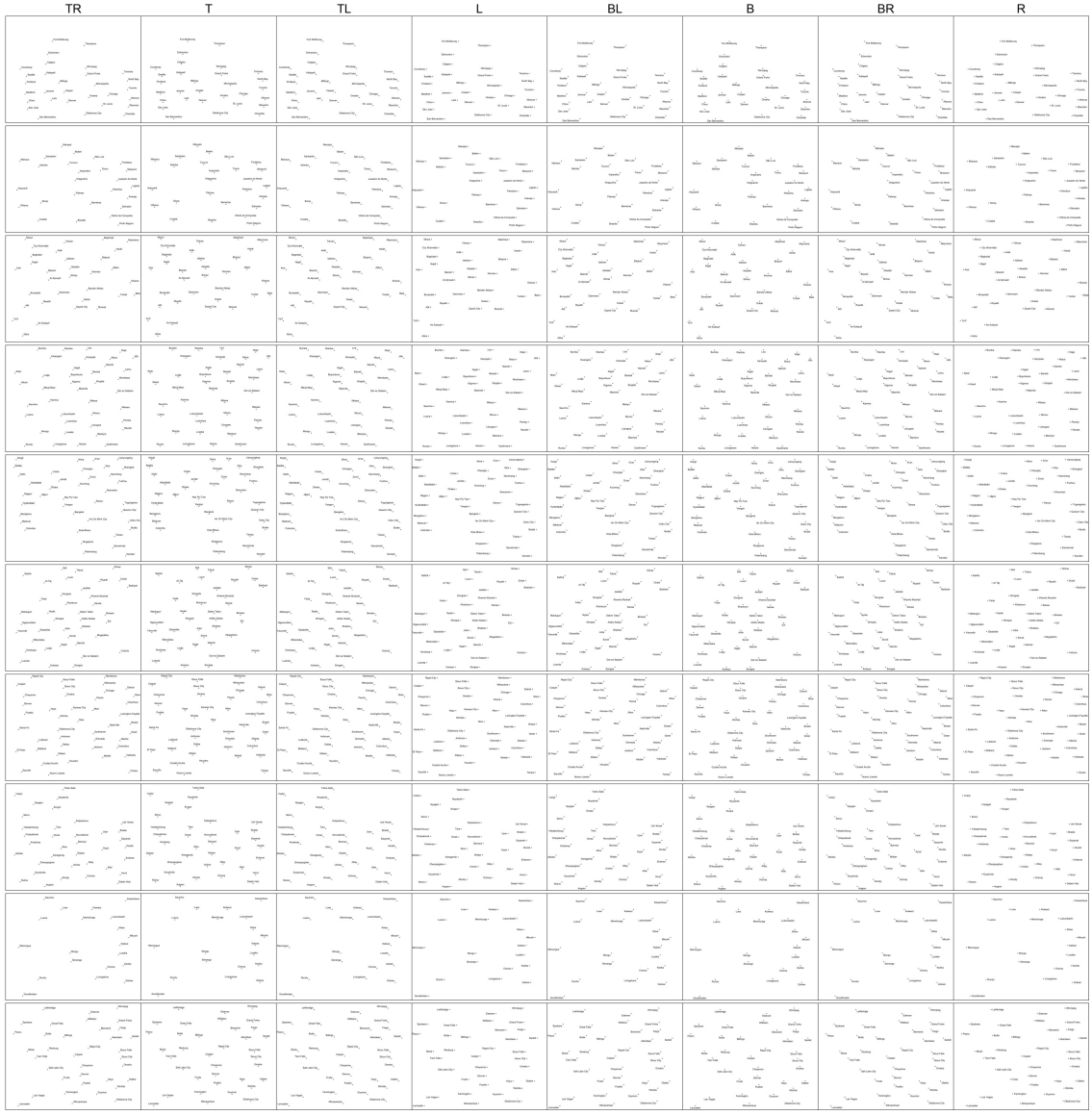


Figure C.17: **`PerceptPPO` Study:** Visualizations of areas 0 through 9 used in the user study.

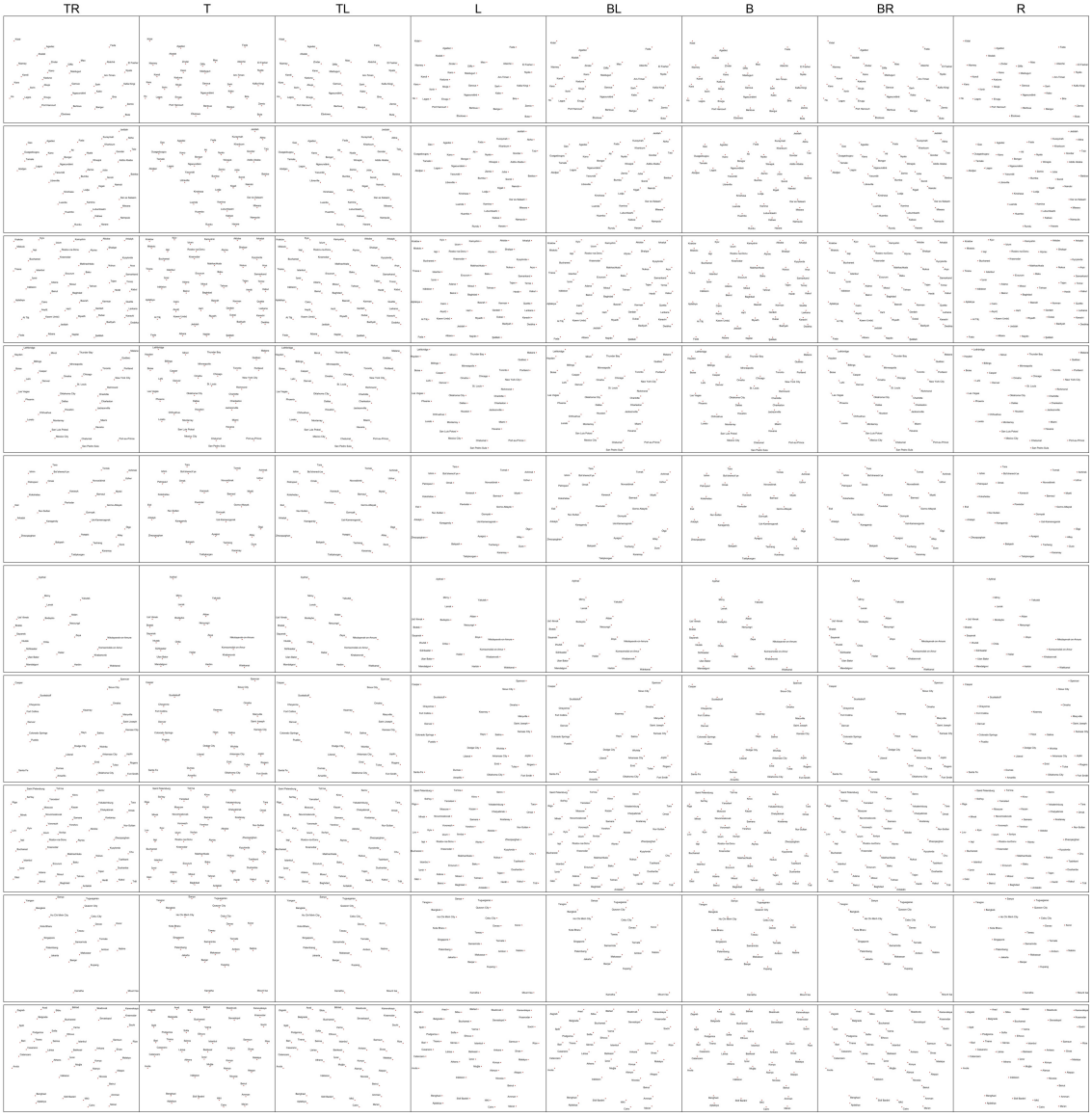


Figure C.18: **`PerceptPPO` Study:** Visualizations of areas 10 through 19 used in the user study.

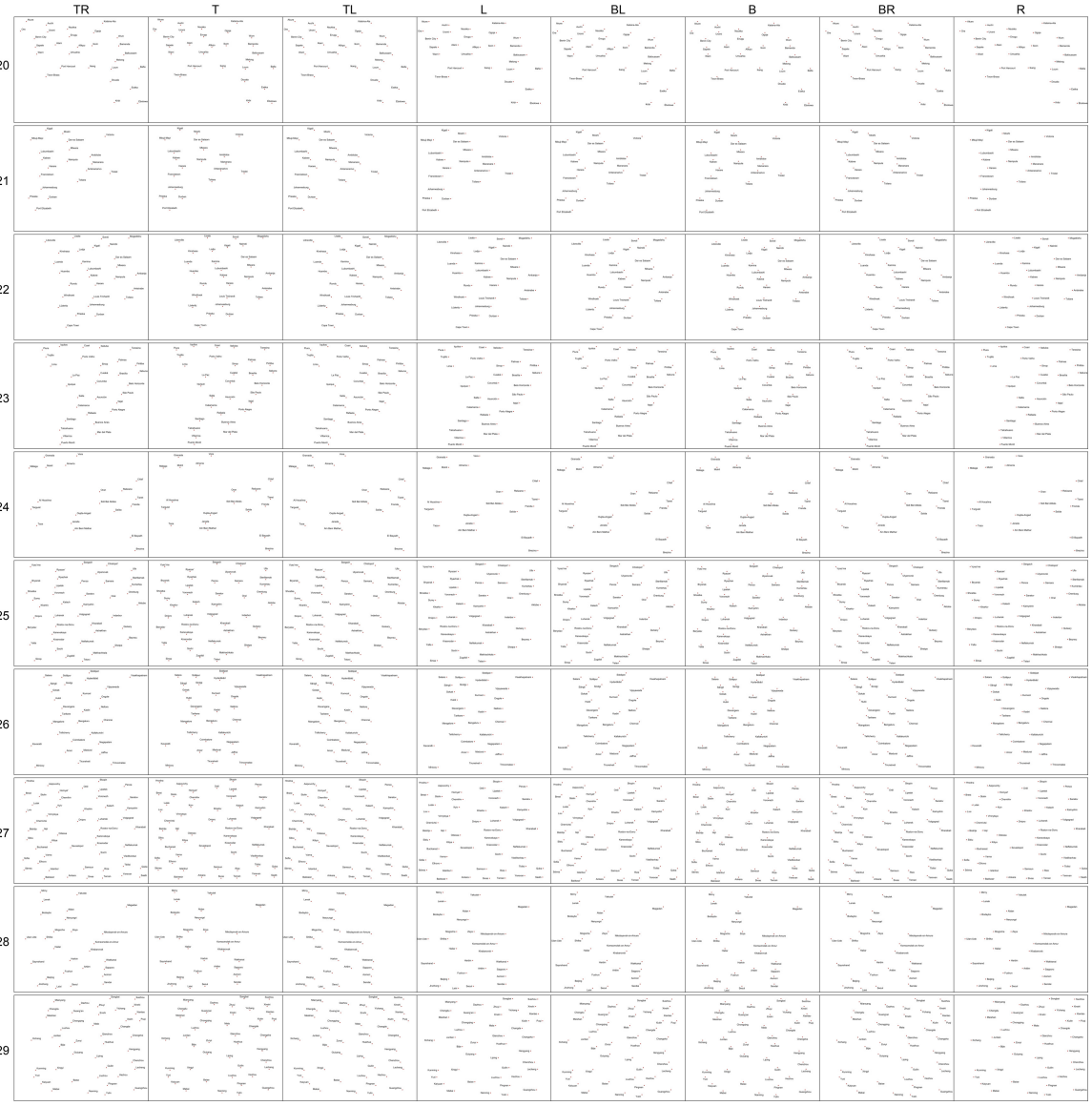


Figure C.19: **`PerceptPPO` Study:** Visualizations of areas 20 through 29 used in the user study.

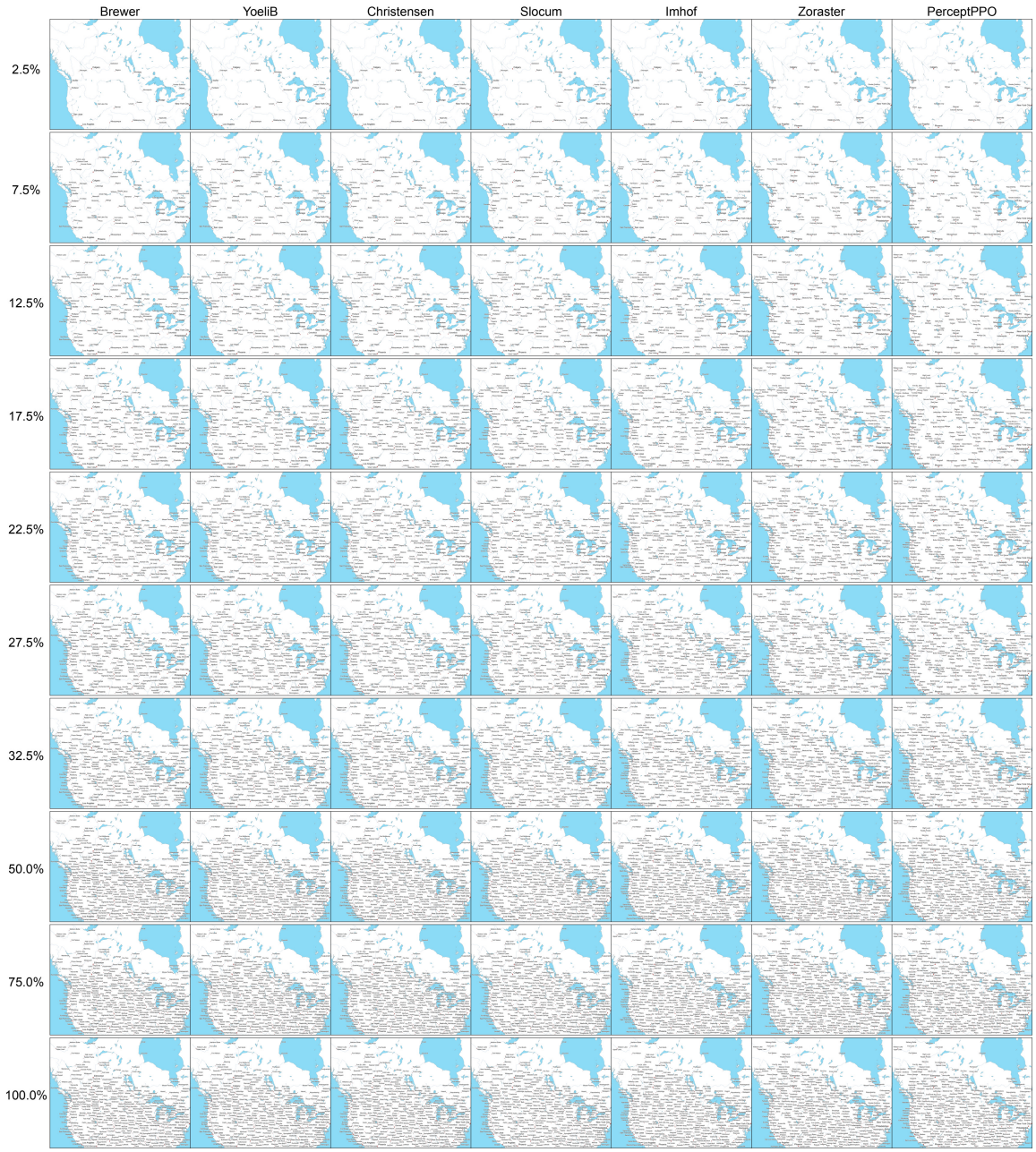


Figure C.20: **Evaluation 1 – Label Density Study:** Visualization of area 0 with various label density thresholds LD_{thr} used in the user study.

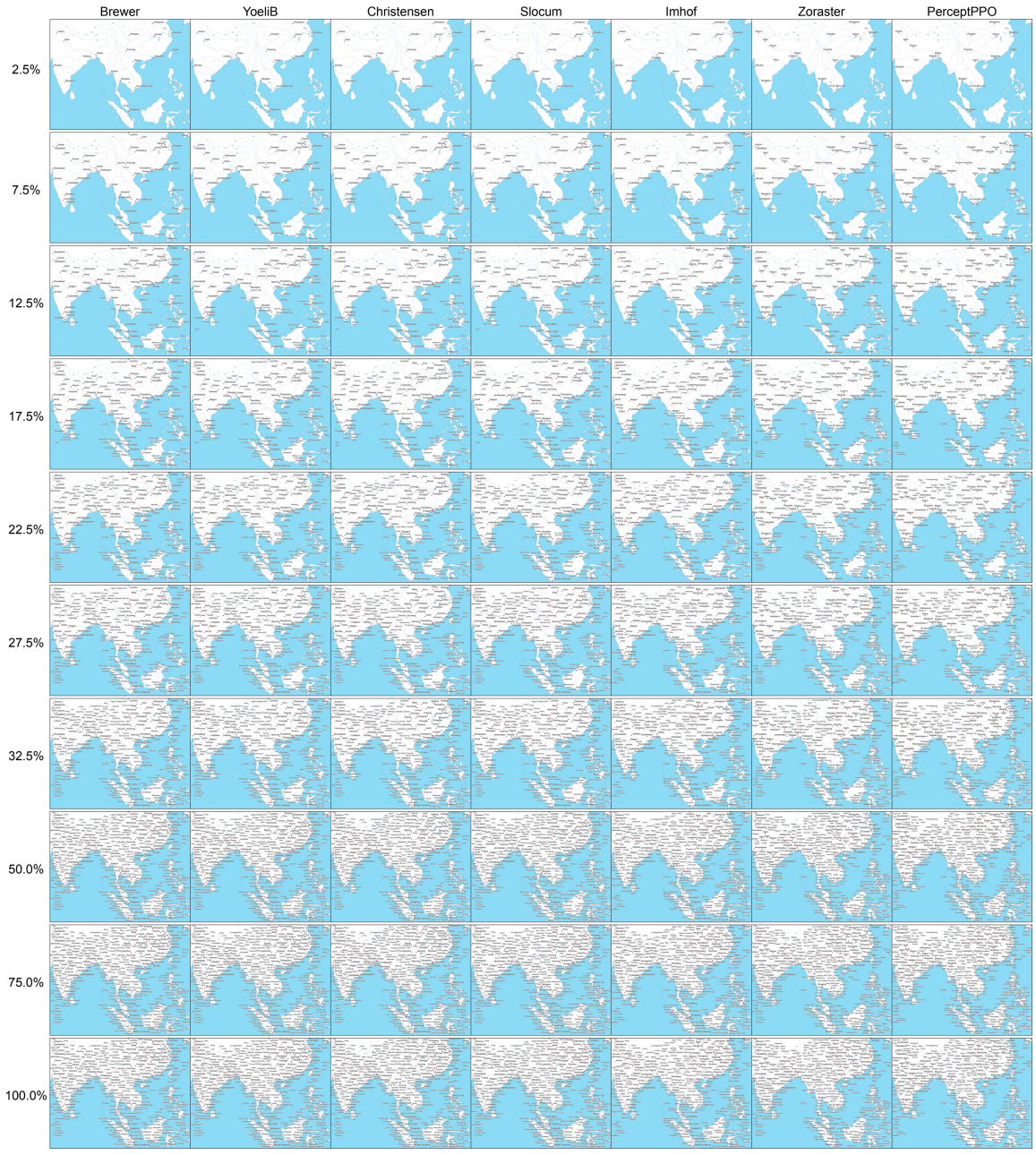


Figure C.21: **Evaluation 1 – Label Density Study:** Visualization of area 4 with various label density thresholds LD_{thr} used in the user study.

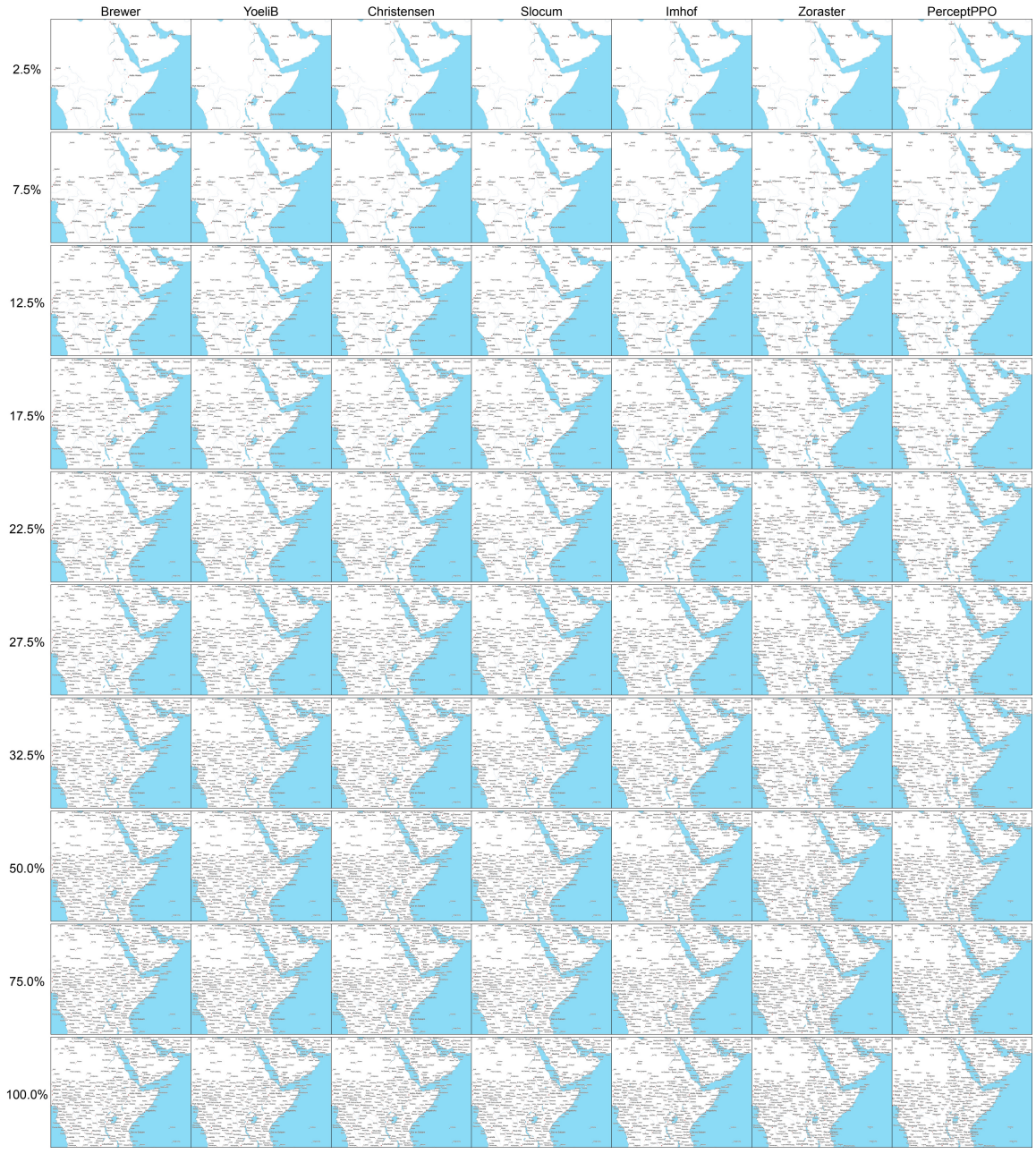


Figure C.22: **Evaluation 1 – Label Density Study:** Visualization of area 5 with various label density thresholds LD_{thr} used in the user study.

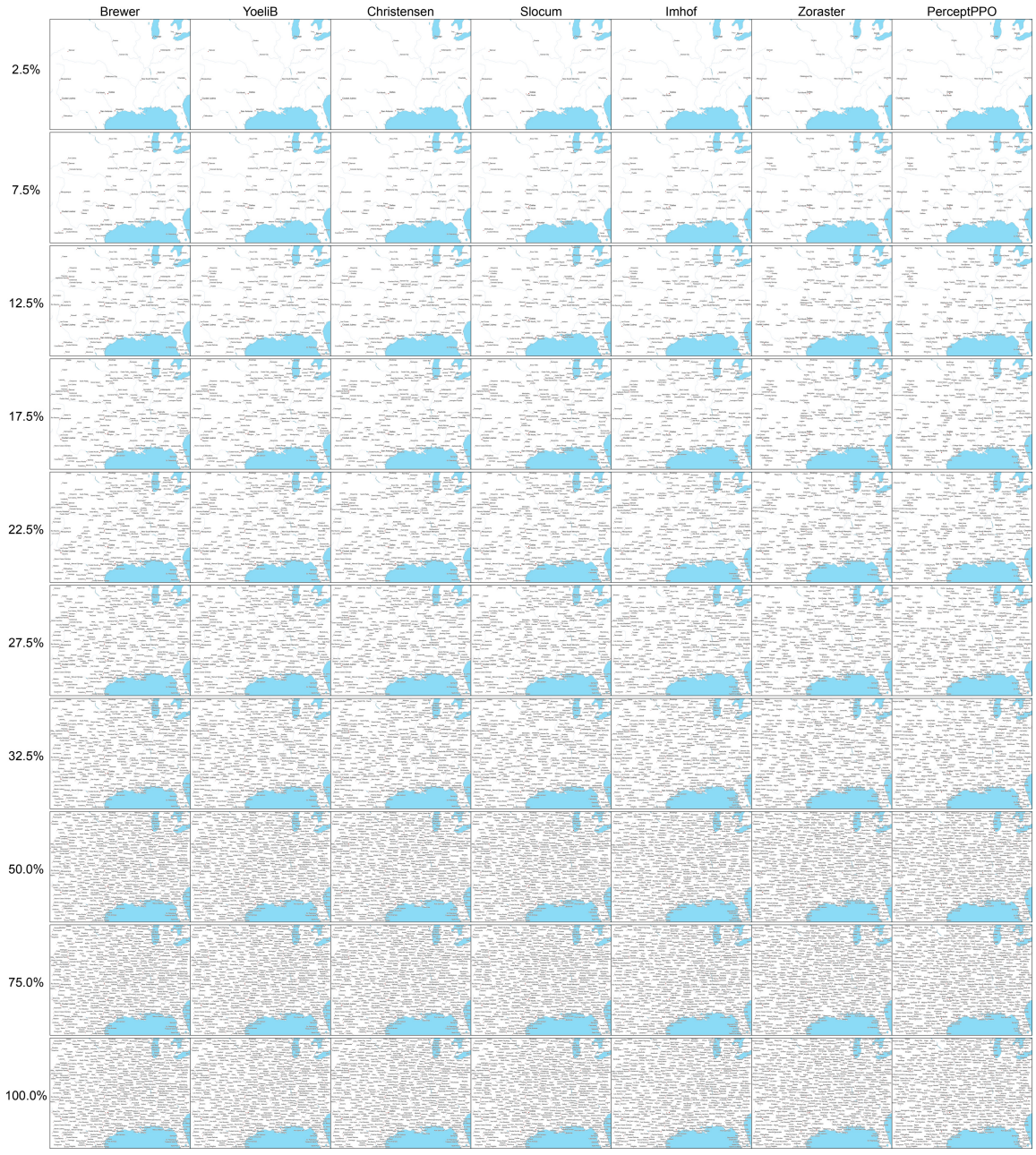


Figure C.23: **Evaluation 1 – Label Density Study:** Visualization of area 6 with various label density thresholds LD_{thr} used in the user study.

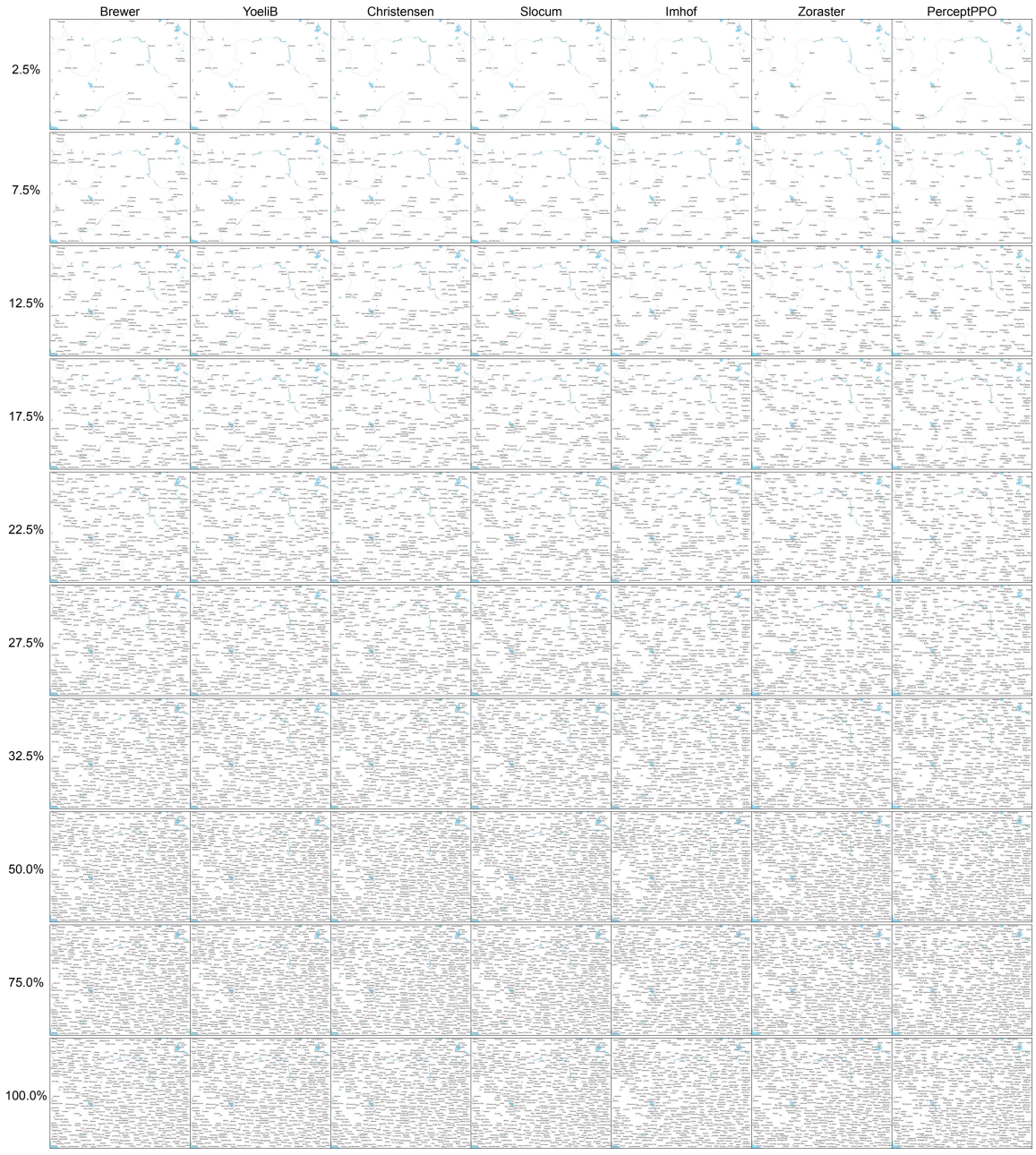


Figure C.24: **Evaluation 1 – Label Density Study:** Visualization of area 9 with various label density thresholds LD_{thr} used in the user study.

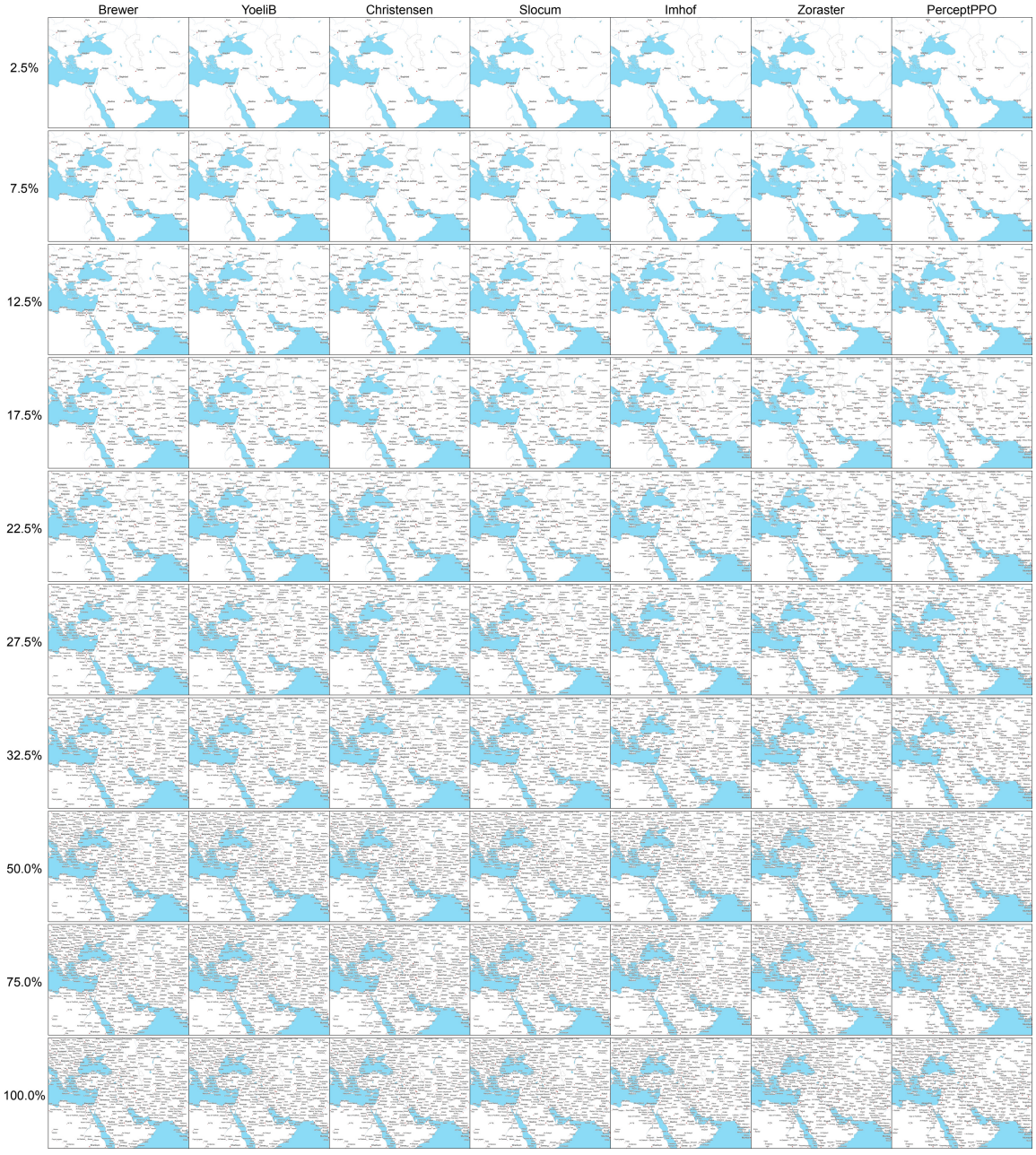


Figure C.25: **Evaluation 1 – Label Density Study:** Visualization of area 12 with various label density thresholds LD_{thr} used in the user study.

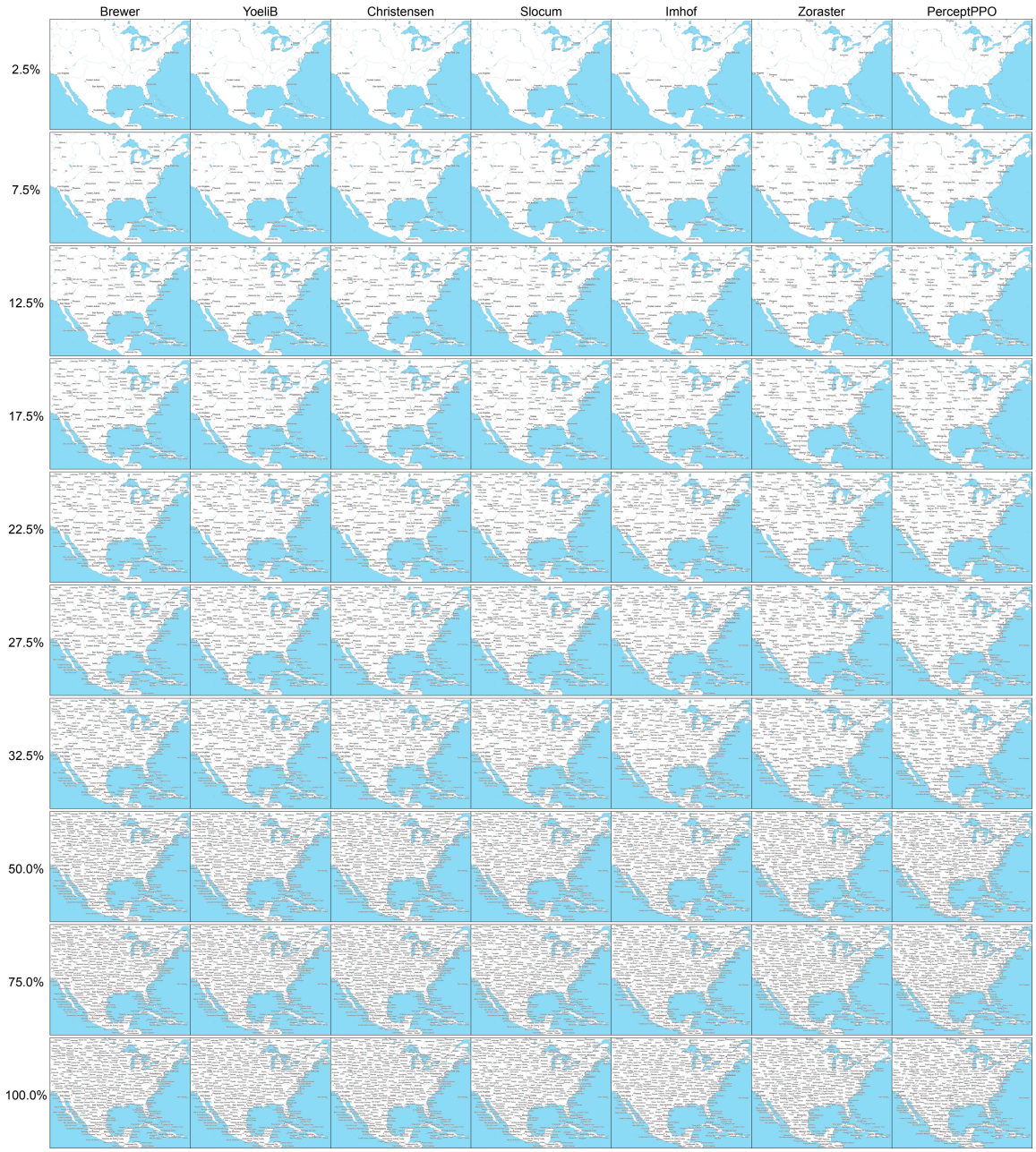


Figure C.26: **Evaluation 1 – Label Density Study:** Visualization of area 13 with various label density thresholds LD_{thr} used in the user study.

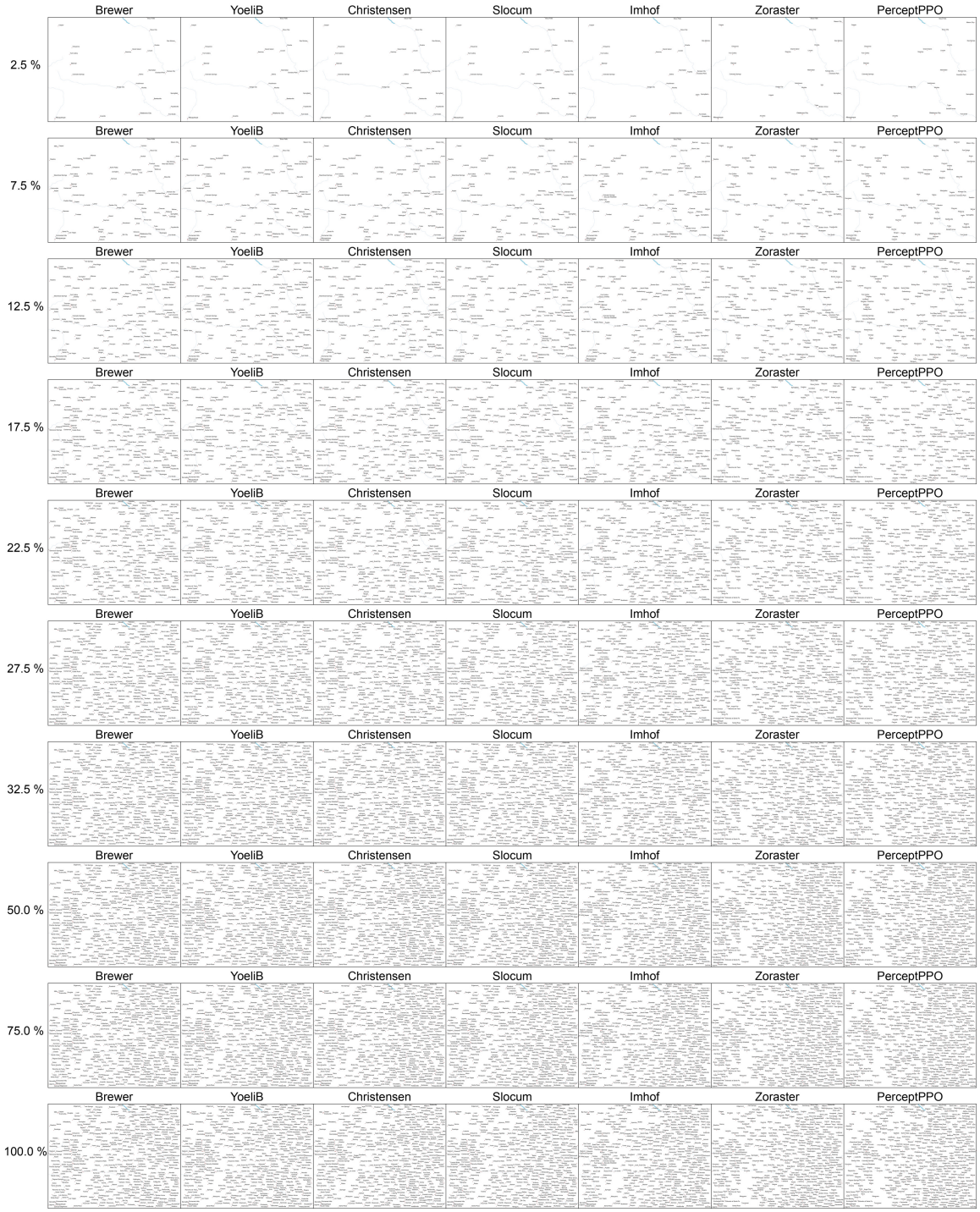


Figure C.27: **Evaluation 1 – Label Density Study:** Visualization of area 16 with various label density thresholds LD_{thr} used in the user study.

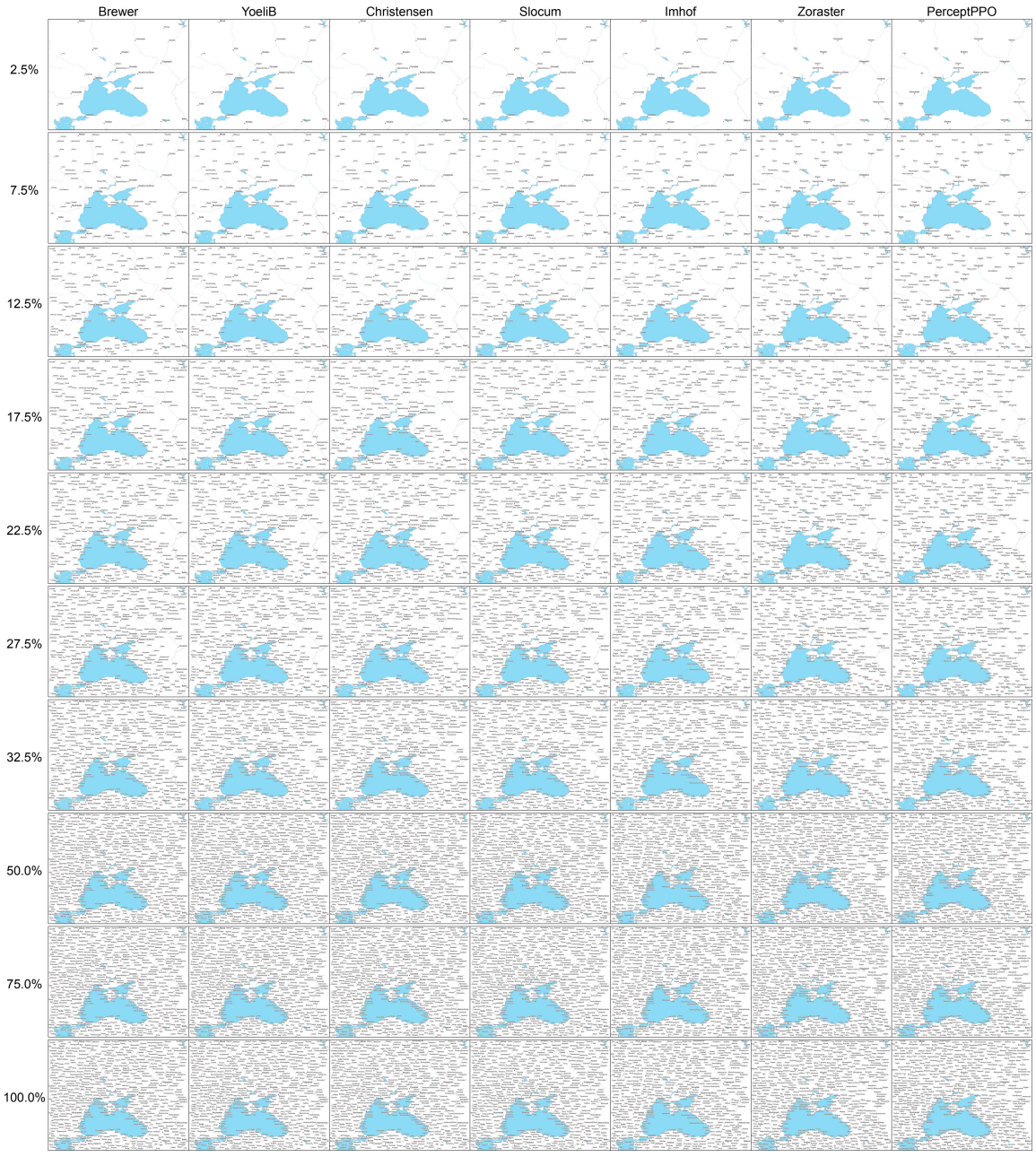


Figure C.28: **Evaluation 1 – Label Density Study:** Visualization of area 27 with various label density thresholds LD_{thr} used in the user study.

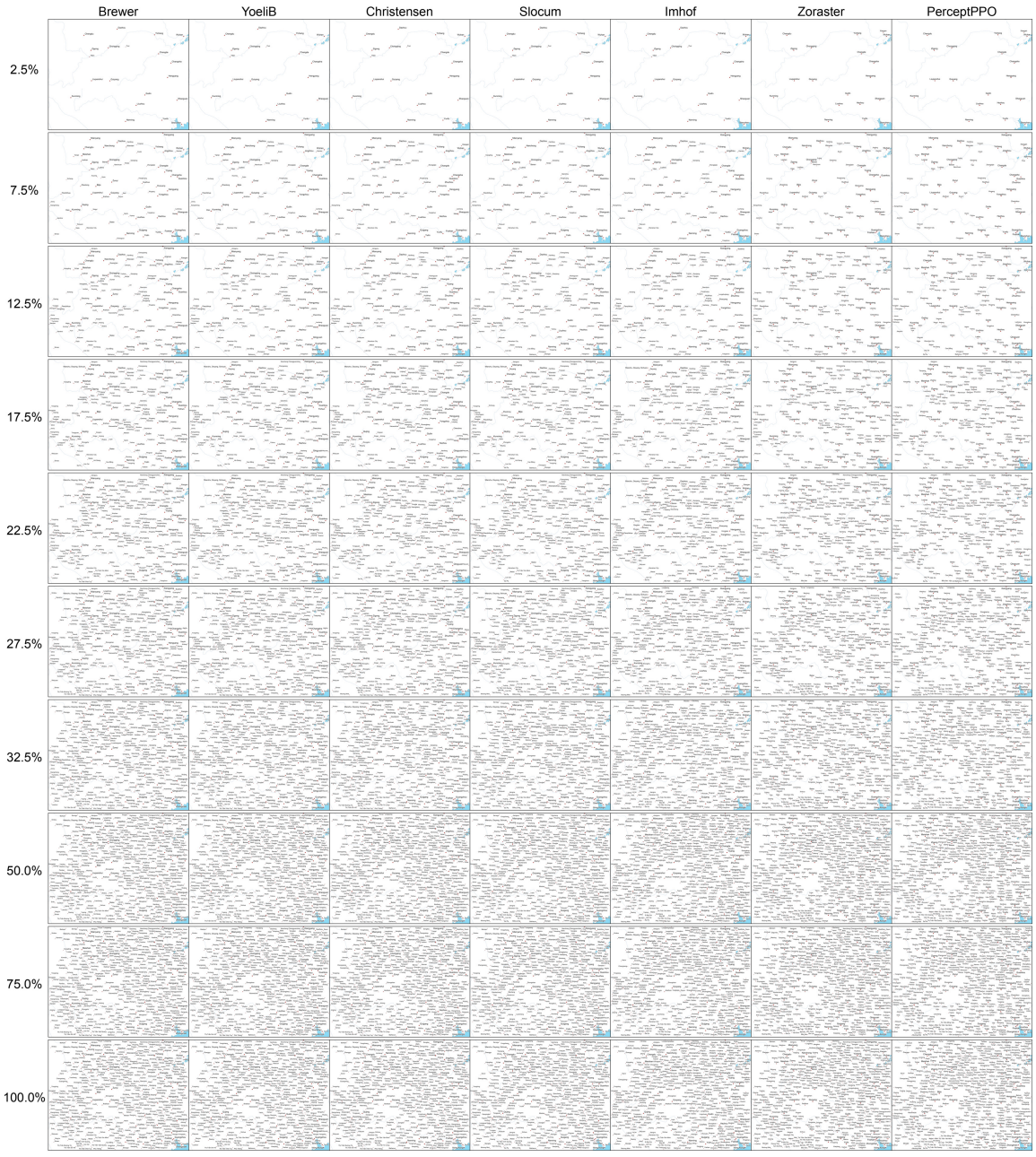


Figure C.29: **Evaluation 1 – Label Density Study:** Visualization of area 29 with various label density thresholds LD_{thr} used in the user study.

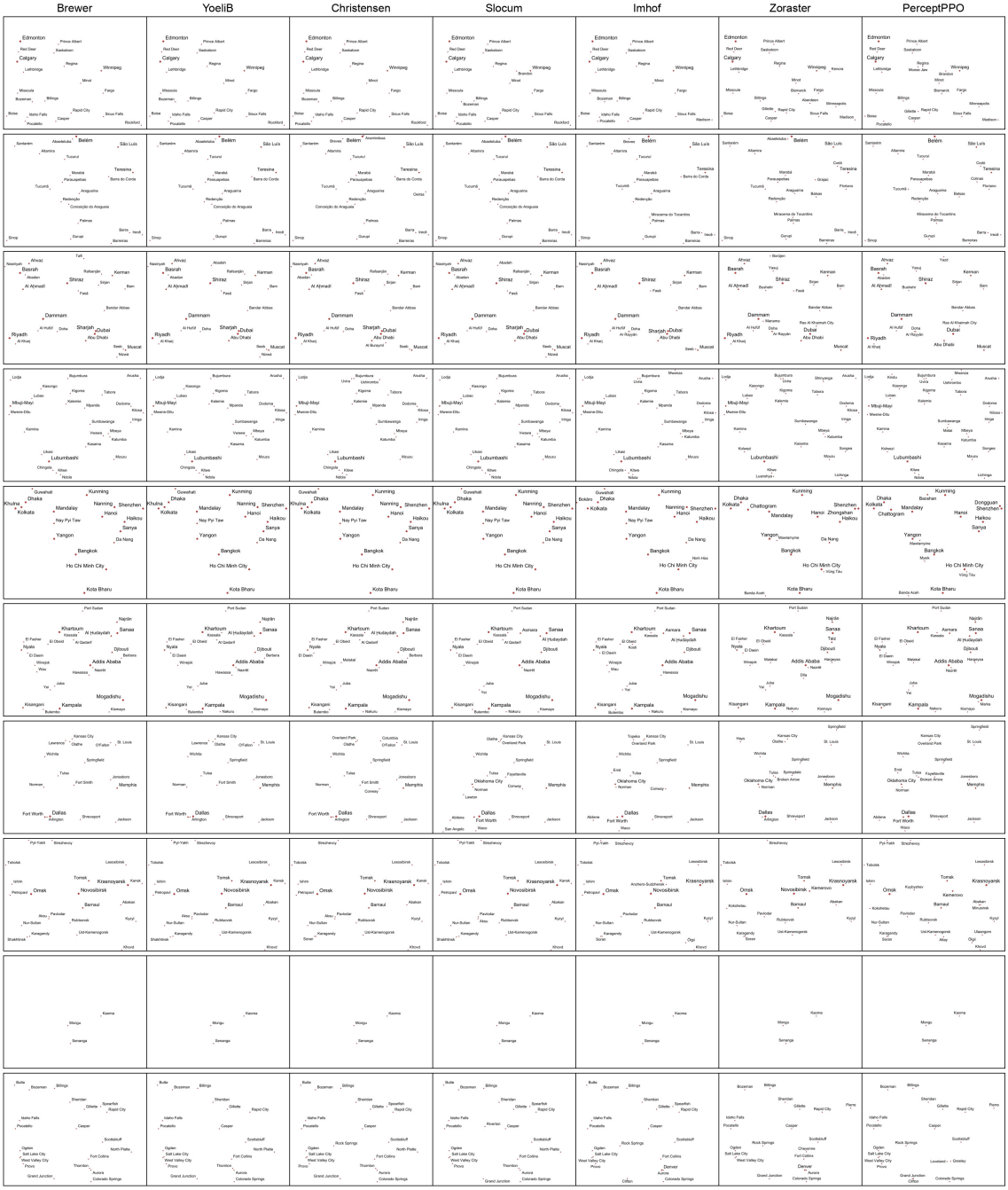


Figure C.30: **Evaluation 2 – Comparison Study of PPOs:** Visualizations of areas 0 through 9 used in the user study.

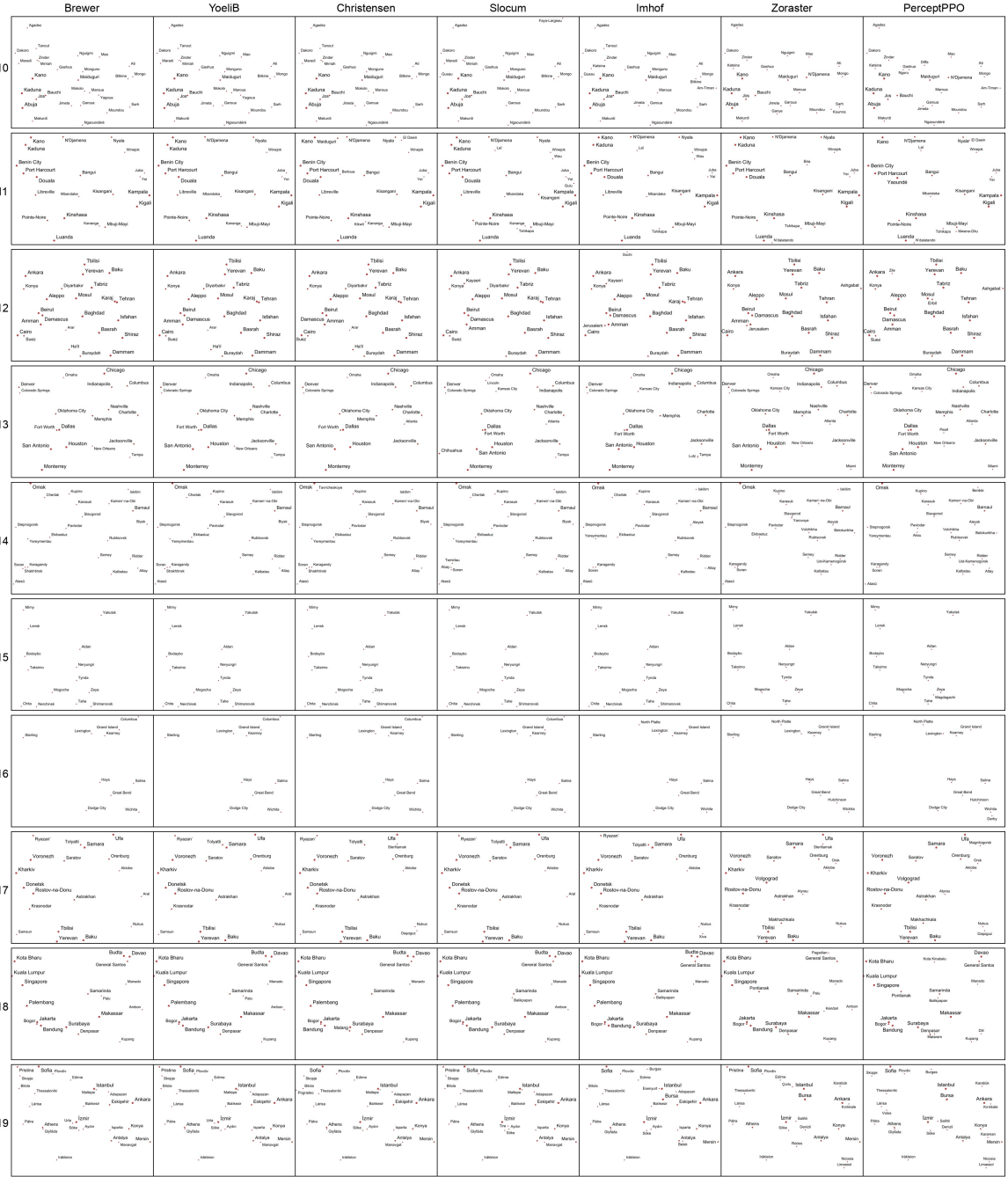


Figure C.31: **Evaluation 2 – Comparison Study of PPOs:** Visualizations of areas 10 through 19 used in the user study.

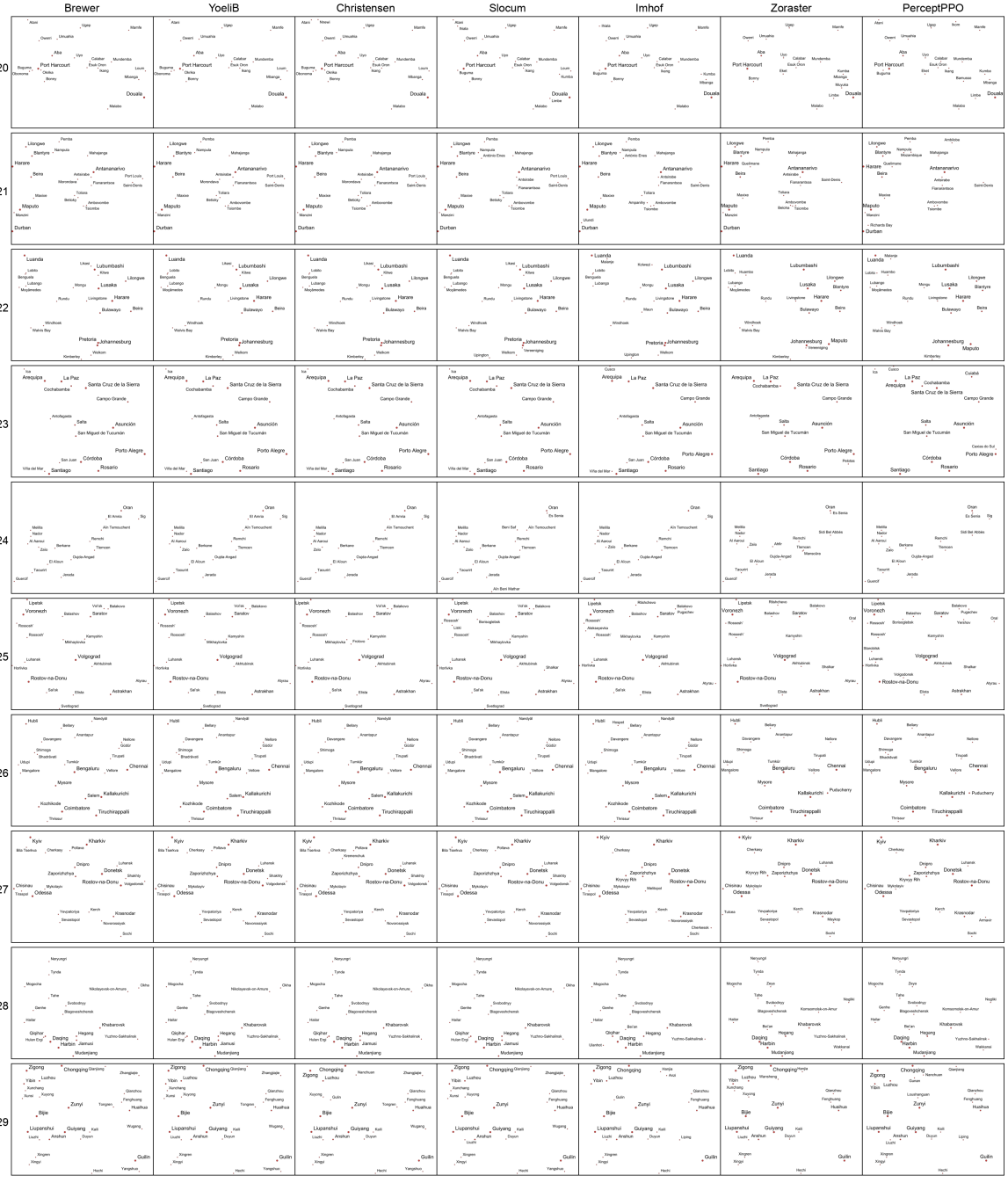


Figure C.32: **Evaluation 2 – Comparison Study of PPOs:** Visualizations of areas 20 through 29 used in the user study.