UM^3: Unsupervised Map to Map Matching

Chaolong Ying
chaolongying@link.cuhk.edu.cn
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China

Yinan Zhang
yinanzhang1@link.cuhk.edu.cn
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China

Lei Zhang
leizhang1@link.cuhk.edu.cn
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China

Jiazhuang Wang
wangjiazhuang@mxnavi.com
MXNavi Co.,Ltd.
Shenyang, Liaoning, China

Shujun Jia*
jiashujun@mxnavi.com
MXNavi Co.,Ltd.
Shenyang, Liaoning, China

Tianshu Yu*
yutianshu@cuhk.edu.cn
The Chinese University of Hong Kong, Shenzhen
Shenzhen, Guangdong, China

arXiv:2508.16874v2 [cs.LG] 18 Jan 2026

Abstract

Map-to-map matching is a critical task for aligning spatial data across heterogeneous sources, yet it remains challenging due to the lack of ground truth correspondences, sparse node features, and scalability demands. In this paper, we propose an unsupervised graph-based framework that addresses these challenges through three key innovations. First, our method is an unsupervised learning approach that requires no training data, which is crucial for large-scale map data where obtaining labeled training samples is challenging. Second, we introduce pseudo coordinates that capture the relative spatial layout of nodes within each map, which enhances feature discriminability and enables scale-invariant learning. Third, we design an mechanism to adaptively balance feature and geometric similarity, as well as a geometric-consistent loss function, ensuring robustness to noisy or incomplete coordinate data. At the implementation level, to handle large-scale maps, we develop a tile-based post-processing pipeline with overlapping regions and majority voting, which enables parallel processing while preserving boundary coherence. Experiments on real-world datasets demonstrate that our method achieves state-of-the-art accuracy in matching tasks, surpassing existing methods by a large margin, particularly in high-noise and large-scale scenarios. Our framework provides a scalable and practical solution for map alignment, offering a robust and efficient alternative to traditional approaches.

1 Introduction

Maps are fundamental tools for understanding and navigating the spatial complexity of the real world. Over the last decade, advancements in geospatial data acquisition and computational power have led to the proliferation of map-related datasets, ranging from street maps and transportation networks to topographic and environmental maps [29]. In this context, the problem of map-to-map matching, which refers to the alignment of maps from different sources or formats to identify similarities and differences between them, has become a critical challenge with far-reaching applications [2, 10]. For instance, map companies often need to align maps from different time periods to track changes in road networks, ensuring that their datasets reflect the most up-to-date geographical information. This process typically involves matching geographic features on one map with corresponding features on another map, enabling the integration and analysis of spatial information.

Accurate map-to-map matching is fundamental for several reasons. One key motivation is the integration of diverse geospatial data sources for monitoring changes in geographic features, such as urban expansion or environmental shifts. For instance, governmental cadastral maps provide highly accurate and authoritative depictions of roads and land boundaries but may lack the frequent updates characteristic of crowd-sourced maps like OpenStreetMap (OSM) [13]. Aligning these complementary datasets allows planners and researchers to combine timely updates with high-precision ground-truth information, benefiting applications such as infrastructure development and urban analytics [12]. Another significant requirement arises in autonomous vehicle navigation. Self-driving systems often depend on high-definition maps for lane-level navigation, which must be reconciled with baseline road maps to ensure accuracy at scale. Conflicting geometries or naming conventions between maps can result in inconsistent path planning or create safety risks [39].

Map-to-map matching, a crucial task in geospatial data integration, faces several key challenges that complicate its implementation. These include data heterogeneity, feature inconsistency, noise-induced errors, computational complexity, and the reliance on manual annotation for labeled data. Maps from different sources often vary in coordinate systems, scales, and feature representations, making direct alignment difficult. Feature inconsistency arises when the same geographic entity is represented differently across maps, such as variations in naming, geometry, or status. Noise from data collection processes, like GPS inaccuracies or digitization errors, further degrades matching accuracy, especially in dense urban environments [32]. The computational complexity of processing large, dynamic datasets with intricate topological relationships demands efficient yet precise algorithms [42, 46]. Additionally, the lack of automated methods for generating high-quality labeled data means that annotations often rely on manual effort, which is time-consuming, costly, and prone to human error. Addressing these challenges necessitates advanced data processing, innovative algorithms, and domain expertise to achieve reliable and scalable map-to-map matching solutions.

Traditional approaches to map-to-map matching have primarily relied on geometric, topological, and rule-based methods to align

*Corresponding authors.

and correlate features across different maps. Geometric methods focus on spatial similarity, using metrics such as Euclidean distance or shape similarity to match corresponding features based on their spatial proximity and geometric properties [8, 34]. While effective in simple scenarios, these methods often struggle with data heterogeneity and noise, as they do not account for contextual or semantic relationships. Topological methods, on the other hand, emphasize the relationships between features, such as connectivity and adjacency, to improve matching accuracy by analyzing the structure of road networks or the arrangement of geographic entities [22]. However, they require robust preprocessing to ensure consistent topological representations across maps, which can be challenging with incomplete or inconsistent data. Rule-based methods incorporate domain knowledge and heuristic rules, such as matching features with similar names or classifications, to guide the matching process [33]. While these methods can improve accuracy in specific contexts, they are often limited by the quality and generality of the rules, making them less adaptable to diverse datasets or dynamic environments. Despite their strengths, traditional approaches face limitations in scalability, robustness, and automation, often requiring significant manual intervention for parameter tuning, rule definition, or error correction [19]. As a result, recent research has increasingly turned to data-driven and machine learning-based methods to address these shortcomings and enhance the performance of map-to-map matching systems.

To the best of our knowledge, there is currently no machine learning-based method specifically designed for map-to-map matching. This gap can be attributed to several significant challenges. First, obtaining large-scale, high-quality map datasets for training is inherently difficult due to the scarcity of publicly available map data and the complexity of integrating heterogeneous map sources. Second, even if such datasets were available, annotating matching information between maps requires extensive domain expertise from professional map engineers, making the labeling process both time-consuming and costly. These challenges collectively render traditional supervised learning approaches impractical for large-scale map-to-map matching applications. To address these limitations, we propose an innovative approach named Unsupervised Map-to-Map Matching (UM^3) that leverages the inherent structural and topological characteristics of maps to achieve accurate matching without the need for manually labeled data. By formulating the problem as an optimization task, our method enables the system to iteratively learn and refine matching patterns, delivering precise alignment results for previously unseen maps in a computationally efficient manner. This approach not only eliminates the dependency on labeled data but also ensures high accuracy and scalability, making it well-suited for real-world geospatial applications. We summarize our contributions as follows:

- **Unsupervised Approach.** We propose a novel unsupervised map-to-map matching method that eliminates the need for manually labeled data, addressing the resource-intensive and time-consuming limitations of traditional supervised learning approaches.
- **Optimization-Based Framework.** By modeling map-to-map matching as an optimization task, our method iteratively learns and refines matching patterns, enabling accurate alignment of large-scale maps without prior knowledge or human intervention.
- **Scalability and Efficiency.** Our approach is designed to handle large-scale geospatial datasets efficiently, providing a scalable solution that achieves high accuracy while significantly reducing computational and manual resource requirements.

2 Related Work

Point Cloud Matching. Point cloud matching focuses on aligning 3D point sets, often captured by LiDAR or other sensors, and has been widely studied in robotics, autonomous driving, and 3D reconstruction. Traditional methods, such as the Iterative Closest Point (ICP) algorithm [6], iteratively minimize the distance between corresponding points but struggle with noise, outliers, and partial overlaps, making it prone to falling into incorrect local optima. An alternative approach to point cloud matching is based on optimal transport theory, such as Sinkhorn Distance (SD) [9]. It treats point clouds as probability distributions, computing an optimal transport plan between these distributions and aiming to minimize the transportation cost. Moreover, the method introduces entropy regularization, which smooths the classical optimal transportation problem. The Sinkhorn-Knopp algorithm [17] enables fast convergence, solving the regularized problem more efficiently. Recent work has explored the integration of point cloud matching into the deep learning framework. One prominent example is PointNetLK [4], which combines the feature extraction capabilities of PointNet [26] with the optimization framework of the Lucas-Kanade [5] optical flow method. PointNetLK directly learns the rigid transformation between point clouds by iteratively refining the alignment, using a learned feature representation to guide the transformation process. Another notable approach is Deep Closest Point (DCP) [36] . It employs PointNet [26] and DGCNN [25] for feature embedding to capture point cloud representations, Transformer for the attention mechanism to refine correspondences, and SVD for computing the rigid transformation, which enables precise alignment of the point clouds.

Graph Matching. Graph matching aims to find correspondences between nodes and edges in two graphs, with applications in computer vision, bioinformatics, and network analysis. Early approaches, such as spectral matching [18], use graph Laplacians to identify correspondences but are sensitive to noise and structural differences. More recent methods incorporate deep learning to extract more dedicated features and optimize matching objectives [30, 43, 45]. Other efforts have been devoted to designing advanced pipelines, such as combining with traditional non-differentiable combinatorial solvers [28], converting the graph matching problem into node classification problem on the association graphs [35], and learning latent topology to enhance matching [44]. While these techniques have shown promise, they often rely on supervised learning and struggle with scalability when applied to large graphs or heterogeneous data.

Trajectory-Map Matching. Trajectory-map matching involves aligning GPS trajectories to a digital road network, a critical task

for navigation, traffic analysis, and location-based services. Early methods, such as geometric approaches [38], match trajectories to the nearest road segments based on spatial proximity but often fail to account for topological constraints. Advanced techniques, like Hidden Markov Models (HMM) [23], incorporate probabilistic frameworks to improve accuracy by considering both spatial and temporal information. More recent work leverages machine learning to enhance matching performance, such as using deep neural networks to learn trajectory-road relationships [20]. They employ spatial-temporal encoders to extract the semantics of trajectories and predict their correspondences with maps during the inference stage [27, 37]. These methods typically require extensive labeled trajectory data and are computationally demanding for large-scale or real-time applications. While trajectory-map matching shares some common ground with map-to-map matching, the two tasks are fundamentally different. In map-to-map matching, the goal is to find correspondences between two maps by aligning nodes and edges based on structural similarity. In contrast, trajectory-map matching involves a dynamic, time-dependent alignment, where the challenge lies in associating spatiotemporal data.

3 Preliminaries

We briefly review the background of this topic in this section, as well as elaborate on the notations.

3.1 Problem Definition

Definition 1. Map. A map M is defined as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with n nodes, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node $v \in \mathcal{V}$ corresponds to a geographical location defined by its latitude and longitude coordinates $\mathbf{x}$, while each edge $e \in \mathcal{E}$ represents a connection between two nodes, typically corresponding to a segment of a road. It is also practical to represent the graph with an adjacency matrix $\mathbf{A} \in \{0,1\}^{n\times n}$.

A map dataset typically consists of a collection of roads, where each road is a sequence of nodes that form a continuous path. Formally, a road R can be defined as an ordered sequence of nodes $R = (v_1, v_2, \ldots, v_k)$, where $v_i \in \mathcal{V}$ for $i = 1, 2, \ldots, k$, and each pair of consecutive nodes (v_i, v_{i+1}) is connected by an edge $e_i \in \mathcal{E}$. Each road is associated with a unique road ID, which serves as an identifier for the road within the map.

Definition 2. Map-to-Map Matching. Given source map M_s and target map M_t, the map-to-map matching problem aims to establish a correspondence between the roads in M_s and M_t such that the alignment maximizes spatial consistency, which refers to ensuring that corresponding roads in the two maps are matched based on their geometric properties and relative positions.

3.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to process data represented as graphs. By leveraging the graph structure, GNNs can capture complex dependencies and interactions, enabling them to learn rich representations of nodes and entire graphs. These models utilize message passing mechanisms to iteratively aggregate information from neighboring nodes, ultimately allowing them to excel in various applications. As GNNs continue to evolve, they offer powerful tools for tackling challenges in machine learning and artificial intelligence across diverse domains. Given an input graph, typical GNNs compute node embeddings $\boldsymbol{h}_u^{(t)}, \forall u \in \mathcal{V}$ with T layers of iterative message passing [40]:

$$\boldsymbol{h}_u^{(t+1)} = \psi\left(\boldsymbol{h}_u^{(t)}, \sum_{v\in\mathcal{N}_u} \boldsymbol{h}_v^{(t)} \cdot \phi(\boldsymbol{e}_{uv})\right) \tag{1}$$

for each $t \in [0, T-1]$, where $\mathcal{N}_u = \{v \in \mathcal{V} | (u,v) \in \mathcal{E}\}$, while ψ and ϕ are neural networks, e.g. implemented using multilayer perceptrons (MLPs).

3.3 Graph Matching

Graph matching is a fundamental problem in computer science and related fields [35, 41, 43], aiming to establish correspondences between the nodes and edges of two graphs. Given two graphs G_1 and G_2, the goal is to find a mapping $f : \mathcal{V}_s \to \mathcal{V}_t$ that maximizes both structural and attribute consistency between the graphs. Recent works often model graph matching as Lawler's Quadratic Assignment Problem (QAP) [21], a well-known combinatorial optimization problem. In this formulation, the goal of graph matching problem is to find a hard correspondence matrix $\mathbf{P} \in \{0,1\}^{n_s\times n_t}$ that maximizes a compatibility function while adhering to row and column constraints. This optimization problem is expressed as:

$$\begin{aligned} \max_{\mathbf{P}} \quad & \mathrm{vec}(\mathbf{P})^\top \mathbf{K} \mathrm{vec}(\mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P} \in \{0,1\}^{n_s\times n_t}, \mathbf{P}\mathbf{1}_{n_t} = \mathbf{1}_{n_s}, \mathbf{P}^\top \mathbf{1}_{n_s} \le \mathbf{1}_{n_t}, \end{aligned} \tag{2}$$

where $\mathbf{K}$ is the affinity matrix derived from the structural information of G_s and G_t and $\mathbf{1}_n$ is a column vector of length n whose elements are all equal to 1. The hard correspondence matrix can be relaxed into a double-stochastic matrix $\mathbf{S} \in [0,1]^{n_s\times n_t}$. In this formulation, each row of $\mathbf{S}$ sums to 1, and each column sums to a value ≤ 1, ensuring that the correspondence probabilities are well-defined.

4 Methodology

4.1 Overview

Map-to-map matching is a challenging problem that seeks to find correspondences between two maps represented as graphs. Unlike traditional graph-graph matching, which often relies on supervised learning with known node-node correspondences as ground truth, our problem setting is fundamentally unsupervised: no ground truth correspondences are available in map data. Moreover, while image-based graph matching benefits from rich node features (e.g., pixel intensities or deep features), map data provides limited discriminability of node features—primarily latitude and longitude coordinates. Combined with the significant variability in road structures and coordinate scales across different regions, it is thus difficult to directly apply existing graph matching techniques.

To address these challenges, we propose an unsupervised map-to-map matching framework that formulates the matching problem as an optimization task. Our approach does not rely on labeled correspondences but instead learns to align maps by optimizing a carefully designed objective function. The key steps of our methodology are as follows: (1) transforming raw latitude and longitude coordinates into pseudo coordinates, a normalized and topology-aware representation that captures the relative spatial structure of

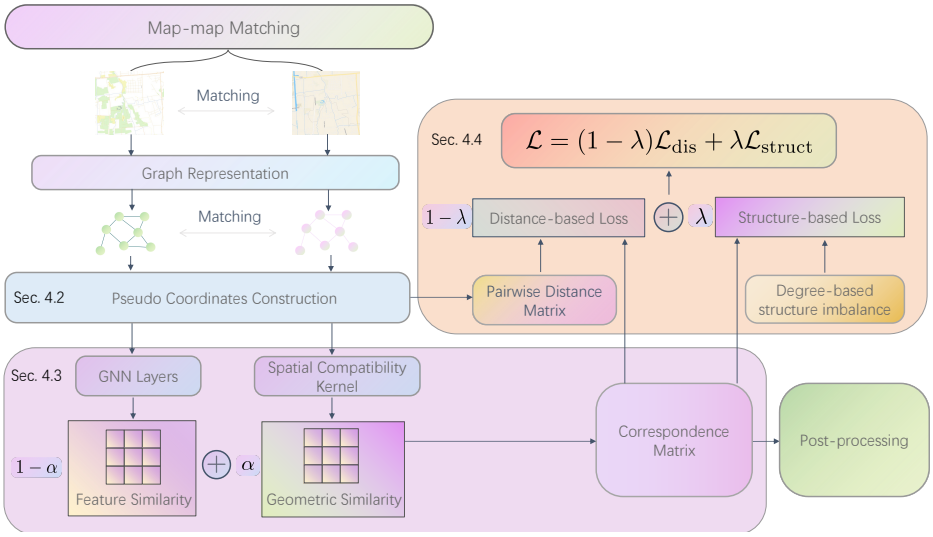


Figure 1: Overview of Unsupervised Map-to-Map Matching. The key steps include: (1) transforming raw GPS coordinates into pseudo coordinates that preserve spatial topology; (2) learning a correspondence matrix through the fusion of feature and geometric similarity; and (3) optimizing a map-to-map matching–specific loss that incorporates spatial and structural constraints. These steps correspond to Sections 4.2, 4.3, and 4.4, respectively.

the map; (2) extracting high-level node embeddings using a GNN $\Psi_\theta(\cdot)$ in Eq. 5 and learning a correspondence matrix that aligns the nodes of the two maps by the fusion of feature similarity and geometric similarity in Eq. 8; and (3) designing a novel loss function tailored to the characteristics of map-to-map matching, which incorporates both spatial and structural constraints to guide the optimization process. An overview of our proposed framework is illustrated in Figure 1. By combining these steps, our method achieves robust and accurate map-to-map matching without the need for supervised training data.

4.2 Pseudo Coordinates Construction

To address the challenges of limited discriminability and the inherent variability in geographical coordinates across different regions, we propose a novel pseudo coordinates construction method that transforms raw latitude and longitude values into the relative spatial layout of nodes within each map. This section details the key steps and motivations behind our approach.

For each node v, we denote its node features as $\mathbf{x} = [x^{(1)}, x^{(2)}] = [\text{lat}(v), \text{lon}(v)]$, where $\text{lat}(\cdot)$ and $\text{lon}(\cdot)$ are the latitude and longitude values of a node. We first compute the minimum and maximum latitude and longitude values across all nodes in the map:

$$\begin{aligned} \text{lat}_{\min} &= \min_{v\in\mathcal{V}} \text{lat}(v), \quad \text{lat}_{\max} = \max_{v\in\mathcal{V}} \text{lat}(v), \\ \text{lon}_{\min} &= \min_{v\in\mathcal{V}} \text{lon}(v), \quad \text{lon}_{\max} = \max_{v\in\mathcal{V}} \text{lon}(v). \end{aligned} \tag{3}$$

These bounds define the spatial envelope of the map and serve as a reference frame for normalization. Then for each node $v \in \mathcal{V}$, we compute its pseudo coordinates $(\bar{x}^{(1)}, \bar{x}^{(2)})$ by normalizing its raw coordinates within the global bounds:

$$\bar{x}^{(1)} = \frac{\text{lat}(v) - \text{lat}_{\min}}{\text{lat}_{\max} - \text{lat}_{\min}}, \qquad \bar{x}^{(2)} = \frac{\text{lon}(v) - \text{lon}_{\min}}{\text{lon}_{\max} - \text{lon}_{\min}} \quad . \tag{4}$$

By normalizing coordinates within the global bounds, the representation becomes agnostic to the absolute scale of the map, enabling consistent learning across regions. The relative positioning of nodes within the map is explicitly encoded, allowing the model to reason about spatial proximity and connectivity. The pseudo coordinates serve as initial node features for GNNs, providing a structured input that aligns with the inductive biases of message-passing architectures.

4.3 Learning Node Correspondence

Given the pseudo coordinates as initial node features, the next step in our framework is to learn a node correspondence matrix that aligns the nodes of two maps. Our approach formulates the map-to-map matching problem as an unsupervised optimization task, where the goal is to find a correspondence matrix that maximizes the similarity between the two graphs. Different from traditional graph matching frameworks that only focus on feature similarity, we propose a geometric-aware similarity fusion mechanism that unifies feature affinity and spatial proximity into a joint optimization framework. This innovation is motivated by the observation that map data inherently exhibits a duality of topological connectivity and geometric continuity, which must be harmonized for robust correspondence learning.

Feature Similarity Learning. As for learning the feature similarity, we follow related approaches [11] to model the node-to-node correspondence by computing pairwise node similarities between the two maps. Specifically, given the initial node coordinate matrix $\mathbf{X}$, the edge feature matrix $\mathbf{E}$ is constructed by computing the great-circle distance between nodes. Together with the pseudo coordinates as initial node features, the latent node embeddings are

computed as

$$\mathbf{H}_s = \Psi_\theta(\mathbf{A}_s, \bar{\mathbf{X}}_s, \mathbf{E}_s) \qquad \mathbf{H}_t = \Psi_\theta(\mathbf{A}_t, \bar{\mathbf{X}}_t, \mathbf{E}_t) \tag{5}$$

through a shared neural network Ψ_θ. Then feature similarity is obtained by

$$\hat{\mathbf{S}} = \mathbf{H}_\mathbf{s}^\top \mathbf{H}_\mathbf{t}. \tag{6}$$

In our implementation, Ψ_θ is a GNN designed to obtain permutation-equivariant node representations [14].

Geometric Similarity Learning. When two maps correspond to the same geographic region, their respective nodes should be spatially close to their true counterparts. In an ideal scenario, a node in one map should not be matched to a node that is far away in the other map. However, due to factors such as structural variations, noise, and incomplete data, erroneous matches may still occur.

To mitigate such errors, we introduce a Geometric Similarity Learning module that incorporates spatial constraints into the matching process. This module ensures that the learned correspondences favor geometrically plausible matches by penalizing associations between nodes that are significantly distant from each other. By integrating this geometric prior into the initial feature based similarity, we improve the robustness of our method against incorrect matches and enhance the overall accuracy of the map alignment process. Specifically, we introduce a spatial compatibility kernel $\exp(-\psi_\theta(\mathbf{D}))$, derived from the pairwise distance matrix $\mathbf{D}$, to refine the initial feature-based similarity matrix. Here ψ_θ is an MLP, and the pairwise distance matrix $\mathbf{D} \in \mathbb{R}^{n_s \times n_t}$ is computed based on the pseudo coordinates for the two maps, where each entry $\mathbf{D}_{uv}$ represents the Euclidean distance between node $u \in \mathcal{V}_s$ and $v \in \mathcal{V}_t$:

$$\mathbf{D}_{uv} = ||\bar{\mathbf{X}}_1[u] - \bar{\mathbf{X}}_2[v]||_2. \tag{7}$$

Here $\bar{\mathbf{X}}_s[u]$, $\bar{\mathbf{X}}_t[v]$ denote the pseudo coordinates of nodes u and v. Finally, the correspondence matrix is learned by the fusion of feature and geometric similarity:

$$\mathbf{S} = \text{Sinkhorn}(\alpha\hat{\mathbf{S}} + (1-\alpha)\exp(-\psi_\theta(\mathbf{D}))), \tag{8}$$

where Sinkhorn$(\cdot)$ is a normalization operator to obtain double-stochastic correspondence matrices [31], α is a **learnable** hyperparameter balancing the contribution of feature similarity and geometric similarity, taking into account that in noisy scenarios, the coordinates may be inaccurate, and thus the spatial compatibility kernel may not provide reliable refinements.

The proposed spatial compatibility kernel acts as a spatial attention gate, adaptively amplifying the similarity scores for node pairs that are both feature-similar and geometrically proximate, while suppressing matches that lack spatial coherence. This mimics the human cognitive process of map alignment, where structural consistency and positional continuity are jointly evaluated.

4.4 Unsupervised Loss Function

To guide the learning of node correspondences in an unsupervised manner, we design a novel loss function that leverages the pairwise distances between node in the two maps. The key idea is to encourage the correspondence matrix $\mathbf{S}$ to align nodes that are not only feature-similar but also structurally consistent in terms of their spatial relationships.

We define $\mathcal{L}_{\text{dis}}$ as the L_2 norm of the element-wise product of the pairwise distance matrix $\mathbf{D}$ and the correspondence matrix $\mathbf{S}$:

$$\mathcal{L}_{\text{dis}} = ||\mathbf{D} \odot \mathbf{S}||_2, \tag{9}$$

where $\odot$ denotes the Hadamard product operator.

In real-world scenarios, the accuracy of coordinate information can be compromised due to noise, measurement errors, or incomplete data. Relying solely on coordinate-based features may lead to suboptimal matching results, especially when the noise level is high. To address this issue, we draw inspiration from the irregularity measure in graph theory [3] to minimize the imbalance between graph pairs, which posits that matched nodes should exhibit balanced structural properties in addition to feature and geometric similarity.

To this end, we propose a structure-based loss term for map-to-map matching task $\mathcal{L}_{\text{struct}}$ that penalizes mismatches in the local structural properties of nodes. Specifically, we compute a structural difference matrix $\mathbf{T} \in \mathbb{R}^{n_s \times n_t}$, where each entry $\mathbf{T}_{uv}$ is computed by

$$\mathbf{T}_{uv} = \text{Deg}(u) - \text{Deg}(v), \tag{10}$$

where $\text{Deg}(u)$ is the normalized degree of node u based on the one-hop neighborhood of a node, defined as:

$$\text{Deg}(u) = \frac{1}{|\mathcal{N}_u| + 1} \sum_{v \in \mathcal{N}_u \cup \{u\}} \text{degree}(v), \tag{11}$$

where $\text{degree}(v)$ is the degree of v. Similarly, $\mathcal{L}_{\text{struct}}$ is defined as

$$\mathcal{L}_{\text{struct}} = ||\mathbf{T} \odot \mathbf{S}||_2. \tag{12}$$

To ensure robust and accurate map-to-map matching, we design a composite loss function that combines both distance-based and structure-based constraints. The overall loss function is defined as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{\text{dis}} + \lambda\mathcal{L}_{\text{struct}}, \tag{13}$$

where λ is a weighting hyperparameter that balances the contributions of the two loss terms.

During the inference phase, the hard assignment is obtained via the Hungarian algorithm [7] as a post processing step, i.e. $\mathbf{P} = \text{Hungarian}(\mathbf{S})$, where $\mathbf{S}$ is computed via Eq. 8.

4.5 Extension to Large-scale Maps

For large-scale maps, processing the entire graph in a single batch is often infeasible due to memory limitations. To address this, we adopt a tile-based processing strategy. The input map is divided into a $k \times k$ grid of overlapping tiles, with adjacent tiles overlapping by a predefined portion of their spatial extent to mitigate information loss at tile boundaries. This overlap ensures that nodes near tile edges are included in multiple tiles, reducing fragmentation of matches across boundaries. Nodes in overlapping regions may receive conflicting correspondence assignments from different tiles. To resolve this, we implement a majority voting scheme, where each node in the overlap region collects all candidate matches from the tiles it belongs to, and the final match is determined by selecting the correspondence with the highest aggregated probability across all overlapping tiles. Ties are resolved by prioritizing matches with smaller spatial distances in the pseudo coordinate space.

Since tiles are processed independently, our framework naturally supports parallel computation. On GPU-enabled systems, multiple tiles can be processed concurrently using batched operations. For distributed systems, tiles can be assigned to different workers, with results aggregated centrally after processing. This design significantly reduces wall-clock time for large-scale maps, ensuring that our method is both practical for real-world applications and scalable to industrial-scale map datasets.

5 Experiments

In this section, we evaluate the performance of our method on map-to-map matching problem. We conduct experiments on real-world datasets collected from three distinct geographic regions to validate its effectiveness in practical scenarios. To assess the robustness of our method under noisy conditions, we generate synthetic datasets by introducing controlled levels of noise to the coordinates and evaluate its performance against baseline methods. To comprehensively assess performance, we compare our method against several baseline approaches in terms of both accuracy and computational efficiency, demonstrating its advantages in large-scale map-matching tasks. Finally, we conduct a series of parameter analyses and ablation studies to evaluate the effectiveness of our method.

5.1 Experimental Setup

Datasets. Our experiments are conducted on map data from three different regions: Boston (USA), Ichikawa (Japan), Shanghai (China), and Bremen (Germany), with statistics in Table 1. The dataset for Boston is obtained from a publicly available source [1], the dataset for Ichikawa is acquired from an official government website [24], the dataset for Shanghai is collected from an industrial partner, and the dataset for Bremen is from OSM [13]. To evaluate the performance in real-world scenarios, we align these datasets with their corresponding regions in the latest OSM data. Specifically, in the Bremen dataset, we perform map-to-map matching between the 2014 and 2025 maps of a selected region, with both maps collected from OSM. Since the source and target maps originate from the same data source, their identifiers are consistent (e.g., unique road IDs in OSM), which makes it straightforward to obtain ground-truth correspondences. Due to differences in data sources, collection methods, and acquisition times, there exist inherent discrepancies between the collected road network data and the corresponding OSM data. These inherent differences make the dataset particularly challenging and realistic for evaluating map-to-map matching algorithms. The variety in geographic regions and data sources ensures the robustness of our method across different road network structures and data acquisition conditions.

To evaluate the robustness of our method under noisy conditions, we create synthetic datasets by introducing artificial noise into the OSM road networks of the Boston, Ichikawa, and Shanghai regions. The noise model is inspired by prior work [23], which suggests that GPS signals can be approximated by Gaussian noise with a mean of 0 and a standard deviation of $\sigma = 4.07$ meters. Based on this model, we define three noise levels to simulate varying degrees of GPS inaccuracies: low noise σ, medium noise 5σ, and high noise 10σ. For each noise level, we independently perturb the latitude and longitude coordinates of every node in the original OSM map by adding Gaussian noise with the corresponding standard deviation. This synthetic dataset allows us to systematically assess our method's performance across different levels of positional uncertainty.

Table 1: The statistics of real-world map-to-map matching datasets.

	#nodes	#edges	area
Boston	2251	2438	2.2km × 1.8km
Ichikawa	2496	2825	2km × 2km
Shanghai	161	178	0.6km × 0.5km
Boston-L	10516	11821	5.2km × 2.7km
Bremen	119445	126999	23.67km × 32.96km

Baseline Methods. As the first method for map-to-map matching, we compare our approach against several relevant baseline methods. Since map data consists primarily of coordinate points, which are conceptually similar to point cloud data, we first compare our method to point cloud matching algorithms, including ICP [6] and SD [9]. Additionally, we convert one of the maps into trajectory data by performing both Depth-First Search (DFS) traversal and Random Walk (RW) to simulate the movement of a vehicle along the roads. Then these sampled segments are fed into a trajectory-map matching method HMM [23] for map-to-map matching. Furthermore, graph matching typically requires node-level supervision for training. To simulate this, we extract another group of data in OSM and introduce low, medium, and high noise perturbations to generate matching pairs, thus creating the necessary training labels for graph matching. We select the Neural Graph Matching (NGM) [35] method as the representative graph matching approach.

Evaluation. For the map-to-map matching problem, we evaluate the performance based on the node matching results, where the correctness of the node correspondences directly influences the accuracy of road correspondences. Specifically, for each matched node, we identify the corresponding road in the other map and check whether the road matches. The Accuracy is calculated as the proportion of correctly matched roads. There are a few special cases in our evaluation to be treated differently. At intersections, a node may belong to multiple roads. In such cases, we consider the match correct if at least one of the corresponding road pairs is correct. Besides, if certain roads from the source map do not have corresponding roads in the target map, the nodes on these roads may be matched incorrectly, but this is not penalized as mismatch in the evaluation. This approach ensures that the evaluation is both realistic and fair, accounting for the complexity of real-world road networks and the inherent ambiguities in map data.

5.2 Implementation Details

Hyperparameters. In our method, we use two neural networks: the first $\mathbf{\Psi}_\theta$ is an 3-layer Graph Convolutional Network (GCN) [16], which utilizes 32 hidden dimensions. The second ψ_θ is a two layer MLP with 32 hidden dimensions. For the Sinkhorn layer, we set the number of iterations to 20 to allow for sufficient optimization. The

model is trained using the Adam [15] optimizer with a learning rate of 0.001 for 200 epoches. These hyperparameter settings were chosen to balance model performance and computational efficiency, allowing for effective training and convergence across the evaluated tasks.

Model Configuration. The experiments are conducted using an AMD EPYC 7542 CPU and a single NVIDIA 3090 GPU. For HMM method, the DFS order simulates the movement of a vehicle along the roads, converting the map data into a trajectory, which is used as a comparison against trajectory matching methods. Ramdom Walk starts from a random initial node, with the vehicle randomly selecting one of its neighboring nodes at each step. The process continues until all the neighbors of a node have been visited by the trajectory. Once this condition is met, the walk selects a new starting point from the unvisited nodes randomly and repeats the process. This continues until all nodes in the graph have been visited and converted into the trajectory.

For the training of NGM, we select a region in Shanghai from the OSM data, covering an area of 68 km $\times$ 66 km. The map is segmented into 60 blocks for generating the training set. Each block is treated as an independent sample, and further divided into a 3×3 grid of overlapping tiles following the approach in Section 4.5. For each tile, we perform node shuffling to introduce structural variation and prevent overfitting to specific spatial layouts. This ensures that the model learns robust representations of road networks rather than memorizing fixed spatial patterns. Following node shuffling, we purturb them with low, medium, and high noise as what we do in our synthetic datasets. The ground truth matching pairs are generated according to the shuffle mapping, providing a supervised training signal for the model. For the GNN backbone, the same 3-layer GCN architecture used in our method is adopted. we train the NGM model for 200 epochs with a learning rate of 0.001 and evaluate it on three real-world datasets. Additionally, for both training and testing stages, we employ the same pseudo coordinates construction, Sinkhorn normalization, and Hungarian matching strategy ensuring a fair and consistent comparison between NGM and our proposed method.

The other baseline methods were adapted from their official source code repositories to ensure consistency with their original implementations.

5.3 Results on Real World Datasets

Experimental results on real-world datasets are summarized in Table 2. Our method consistently outperforms all baseline algorithms on the three datasets, achieving significant improvements in matching accuracy. This highlights the effectiveness of our approach in addressing the challenges of map-to-map matching. Meanwhile, we record the running time in Table 4, which shows that our method demonstrates competitive runtime performance, with execution times comparable to the fastest baseline methods. The combination of high accuracy and low runtime makes our method suitable for real-world, large-scale applications.

Point cloud matching methods perform better than other counterparts due to their emphasis on geometric spatial relationships.

Table 2: Results of matching Accuracy (%) ↑ on real-world datasets.

Dataset	Boston	Ichikawa	Shanghai	Bremen
ICP	94.69	82.13	84.47	67.37
SD	94.94	81.46	79.50	67.12
HMM + DFS	81.16	78.76	61.49	27.32
HMM + RW	37.41	60.92	48.45	23.15
NGM	85.96	68.83	80.12	44.85
UM^3	**97.38**	**87.60**	**91.82**	**87.52**

However, they overlook crucial graph structures like road connectivity. Our method outperforms existing approaches by integrating both geometric and structural information. While supervised graph matching techniques like NGM require extensive training and struggle with regional variations, our unsupervised approach offers superior adaptability without requiring labeled data.

The visualization results are presented in Figure 2. Since the map-to-map matching problem involves matching roads to other corresponding roads, we highlight the correctly matched roads. The roads that are either incorrectly matched or have no corresponding match in the ground truth are shown without highlighting. Each road is randomly assigned a color, and matching roads between the two maps are displayed using the same color to indicate their correspondence.

On Large-scale Maps. The Bremen dataset is considerably larger than the other datasets, which introduces more significant challenges for the map-to-map matching task. To this end, we adopt the tile-based strategy described in Section 4.5 to fit the data on our device. Besides, although the source and target maps in this dataset come from the same location and the same source, significant differences have emerged over more than a decade of changes, making it still a challenging task to match the two maps. In particular, many roads present in the source maps no longer have corresponding counterparts in the target maps. These discrepancies are largely due to urban development, road reconstruction, and changes in mapping policies over time. From Table 2, we can observe that our method is still much more effective than the compared baselines. Due to the massive size of the maps, we visualize their overall structures and provide a zoomed-in view of a portion of the matching results as an illustrative example, as shown in Figure 3. The results indicate that our method remains effective in identifying regions with minimal changes and achieves high matching accuracy in these areas, largely due to the integration of both feature similarity and geometric similarity in our matching framework.

On Maps with No Known Ground Truth. We select a large map Boston-L and perform map matching against OSM data. Since the source and target maps come from different sources, it is not possible to directly obtain one-to-one correspondences as ground truth. Moreover, due to the large scale of the data, manual annotation is impractical, and thus we are unable to directly evaluate the performance of our method. Nevertheless, the quality of the map matching results can still be roughly assessed through visual inspection. As shown in Figure 4, the map matching results for

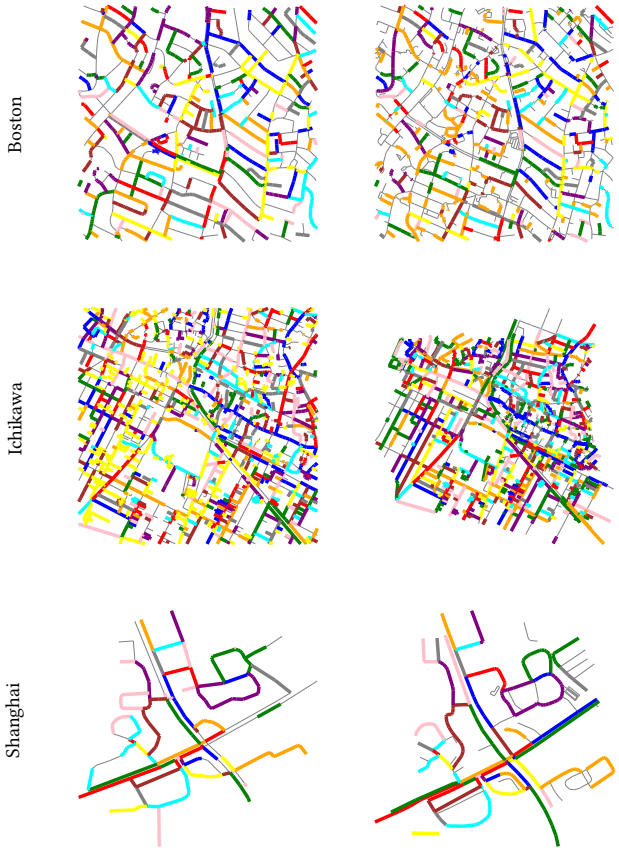


Figure 2: Visualization results of our method on real world datasets. Since the source and target maps are from the same region, the corresponding roads generally maintain similar relative positions across maps. The correctly matched roads are highlighted, while roads shown in thin black lines indicate unmatched or mismatched segments. Here, *unmatched* refers to roads that have no corresponding counterparts in the other map (e.g., they do not exist in the other map).

the large-scale map remain highly accurate, further validating the scalability and reliability of our approach.

5.4 Results on Synthetic Datasets

To simulate real-world scenarios where map data can be highly complex, sourced from diverse origins, and often inaccurate, we evaluated our method and baselines on synthetic datasets with varying levels of noise. The results are presented in Table 3. Overall, our method demonstrates remarkable robustness to noise: it remains completely unaffected under low noise levels, nearly unaffected under medium noise, and only slightly impacted under high noise. This robustness highlights the practical applicability of our method in real-world scenarios, where data quality is often uncertain.

In contrast, the performance of other baseline methods degrades significantly even under low noise conditions. Under high noise

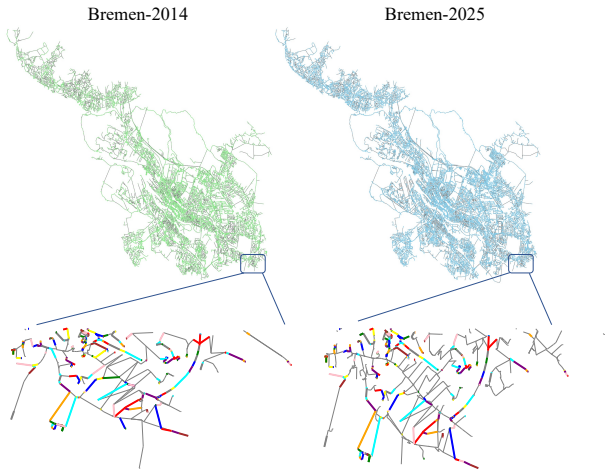


Figure 3: Node matching result of our method on Bremen dataset. The source and target maps were collected in 2014 and 2025, respectively. A zoomed-in view of a portion of the matching results is provided as an illustrative example.

levels, these methods almost entirely fail to provide reliable map-to-map matching results. Notably, the supervised learning method NGM, which was trained with extensive noisy data to enhance its robustness, performs more stably than other baselines. However, it still falls far short of the performance achieved by our method. These results underscore the superiority of our approach in handling noisy and imperfect data, further validating its potential for real-world applications.

5.5 Runtime Comparisons

The runtime comparisons are shown in Table 4. In addition to achieving the best matching accuracy, our method demonstrates competitive runtime, with execution times comparable to the fastest baseline methods. This efficiency is achieved through our unsupervised learning framework, which iteratively refines the matching results without requiring expensive training or extensive computational resources. The combination of high accuracy and low runtime makes our method suitable for real-world, large-scale applications.

5.6 Analysis

Parameter Analysis. Our method involves two key hyperparameters α and λ. The parameter α controls the contribution of feature similarity and geometric similarity, while λ balances the trade-off between $\mathcal{L}_{dis}$ and $\mathcal{L}_{struct}$. In this section, we analyze the impact of these parameters on our model using the Shanghai dataset as an example.

As shown in Figure 5(a), we evaluate the performance of our model with different values of α. Notably, when $\alpha = 0$, the model relies entirely on geometric similarity; when $\alpha = 1$, the model relies entirely on feature similarity. On the selected datasets, where map coordinates are relatively accurate, geometric similarity can complement feature similarity by providing additional spatial information that feature-based methods might miss. As a result, our model achieves strong performance when α is small. On noisy

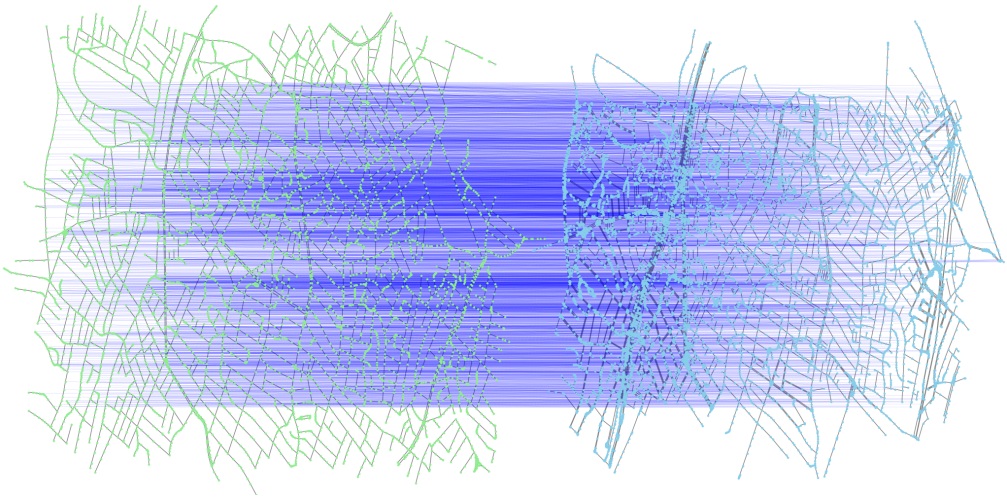
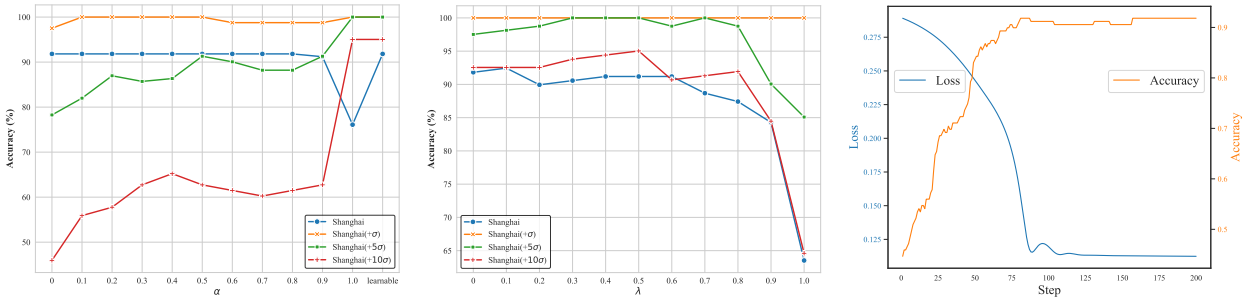


Figure 4: Node matching result of our method on Boston-L dataset. Each light blue line connects a pair of matched nodes from the two maps. Zoom in to better view.



(a) Varying α (b) Varying λ (c) Convergence analysis

Figure 5: Parameter analysis on the Shanghai dataset. (a) and (b) show the impact of varying parameters α and λ, respectively. The x-axis in both plots corresponds to different parameter values, and the y-axis denotes the matching accuracy. (c) illustrates the training dynamics, showing how the loss decreases and the accuracy improves over training steps.

Table 3: Results of matching Accuracy (%) ↑ on synthetic datasets. We consider low, medium, and high noise scales to simulate real world noisy scenarios.

	Low noise (σ)			Medium noise (5σ)			High noise (10σ)		
	Boston	Ichikawa	Shanghai	Boston	Ichikawa	Shanghai	Boston	Ichikawa	Shanghai
ICP	87.94	79.25	83.54	49.54	41.51	39.63	31.22	26.40	24.39
SD	87.88	79.12	76.52	50.83	41.88	39.02	32.31	25.86	20.73
HMM + DFS	0.58	0.54	2.74	1.25	0.33	2.74	0.17	3.39	6.71
HMM + RW	1.12	0.54	2.13	1.12	1.84	3.66	1.08	0.38	3.66
NGM	85.96	68.49	81.37	85.52	67.53	78.88	84.10	66.61	72.67
UM^3	**100**	**100**	**100**	**99.82**	**99.76**	**100**	**95.82**	**97.64**	**95.03**

datasets, however, As shown in the results, the learnable α adaptively learns an optimal balance between feature and geometric similarity, achieving the best performance. In practical applications, users can also manually adjust α based on their prior knowledge or specific requirements.

In Figure 5(b), we evaluate the performance of our model with different values of λ. The parameter λ balances the contributions of two proposed loss terms. A well-chosen λ ensures that the model effectively leverages both geometric and structural information.

Table 4: Runtime comparisons (in seconds) on real world map-to-map matching datasets.

Dataset	Boston	Ichikawa	Shanghai	Bremen
ICP	15.03	15.82	6.03	7.26×10^2
SD	5.37	5.49	0.67	1.31×10^2
HMM + DFS	1.77×10^2	1.34×10^2	12.23	3.89×10^4
HMM + RW	1.12×10^3	9.02×10^2	37.69	5.01×10^4
NGM	2.03×10^4	2.03×10^4	2.03×10^4	2.33×10^4
UM^3	9.92	8.59	3.56	5.17×10^2

When coordinate data is accurate, the computed distances are reliable, and $\mathcal{L}_{dis}$ provides strong guidance for the model to learn better matching relationships. However, in the presence of significant noise, distance information becomes less accurate, and $\mathcal{L}_{struct}$ offers stable support by leveraging structural consistency. This demonstrates that an appropriate balance between these two losses is critical for achieving robust and accurate matching results.

Convergence Analysis. To further analyze the convergence behavior of our method, we visualize the changes in both loss and accuracy throughout the training process, using the Shanghai dataset as a representative example. As shown in Figure 5(c), the loss decreases steadily while the accuracy consistently improves, indicating that the model is able to effectively learn meaningful correspondences over time. This trend validates the design of our proposed loss function, which serves as a reliable surrogate objective for the unsupervised map matching task. The close alignment between the loss reduction and accuracy improvement also demonstrates that the loss captures key aspects of matching quality. Moreover, the method exhibits rapid convergence within a relatively small number of training steps, underscoring both its optimization efficiency and practical effectiveness.

5.7 Ablation Study

To demonstrate the significance of the pseudo coordinates module, we conduct an ablation study by removing this component and directly using raw latitude and longitude as initial features. We name this variant as No Pseudo Coordinates (-NPC). The experimental results in Table 5 show a significant drop in performance when pseudo coordinates are omitted. This is because raw latitude and longitude coordinates within a small region exhibit minimal variation, lacking the discriminative power needed for effective matching. Since both our loss function and distance matrix rely on coordinate information for computation, the absence of meaningful pseudo coordinates leads to large matching errors.

Another part of the ablation study focuses on removing the fusion of feature and geometric similarity by using only one of them to model the node correspondence. This is achieved by setting $\alpha = 0$ or 1. Similarly, the ablation of the structure-based loss is performed by setting $\lambda = 0$. These variants have already been analyzed in Section 5.6, which demonstrates the effectiveness of our proposed modules in addressing the map-to-map matching problem.

Table 5: Results of Accuracy (%) ↑ about the ablation study of our proposed pseudo coordinates.

	UM^3-NPC	UM^3
Shanghai	31.45	91.82
Shanghai $+\sigma$	98.76	100
Shanghai $+5\sigma$	80.75	100
Shanghai $+10\sigma$	49.69	95.03

This experiment highlights the critical role of pseudo coordinates as a foundational component of our method. By generating meaningful and discriminative coordinate representations, the pseudo coordinates module effectively enhances the model's learning capability, ensuring robust and accurate map-to-map matching.

6 Conclusion

In this paper, we introduce the first unsupervised map-to-map matching method, addressing the challenge of aligning maps from different sources. Unlike traditional point cloud matching or trajectory-map matching approaches, our method explicitly models the structural relationships in maps while incorporating geometric similarity learning to improve robustness against noise. We evaluate our approach on several real-world and synthetic datasets from different regions, and compare it against competitive baselines. The results demonstrate that our method is both effective and efficient, especially in noisy scenarios. Our method also exhibits high scalability due to our designed tile-based multi-processing strategies.

Our work establishes a strong foundation for future research in map-to-map matching and opens up new possibilities for applications in map alignment, navigation, and geographic data integration. It demonstrates the feasibility of learning accurate correspondences without requiring manual annotations. In future work, we aim to explore more advanced similarity measures and investigate how our method can be extended to handle dynamic changes in road networks over time.

References

[1] [n. d.]. https://bostonopendata-boston.opendata.arcgis.com/.

[2] Jordi Aguilar, Kevin Buchin, Maike Buchin, Erfan Hosseini Sereshgi, Rodrigo I Silveira, and Carola Wenk. 2024. Graph sampling for map comparison. *ACM Transactions on Spatial Algorithms and Systems* 10, 3 (2024), 1–24.

[3] Michael O Albertson. 1997. The irregularity of a graph. *Ars Combinatoria* 46 (1997), 219–226.

[4] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. 2019. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7163–7172.

[5] Simon Baker and Iain Matthews. 2004. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision* 56 (2004), 221–255.

[6] Paul J Besl and Neil D McKay. 1992. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, Vol. 1611. Spie, 586–606.

[7] Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. 2012. *Assignment problems: revised reprint*. SIAM.

[8] Alireza Chehreghan and Rahim Ali Abbaspour. 2018. A geometric-based approach for road matching on multi-scale datasets using a genetic algorithm. *Cartography and Geographic Information Science* 45, 3 (2018), 255–269.

[9] Marco Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* 26 (2013).

[10] Evgeniy Faerman, Otto Voggenreiter, Felix Borutta, Tobias Emrich, Max Berrendorf, and Matthias Schubert. 2019. Graph alignment networks with node matching scores. *Proceedings of Advances in Neural Information Processing Systems (NIPS)* 2 (2019).

[11] Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. 2020. Deep graph matching consensus. In *ICLR*.

[12] Michael F Goodchild and Linna Li. 2012. Assuring the quality of volunteered geographic information. *Spatial statistics* 1 (2012), 110–120.

[13] Mordechai Haklay and Patrick Weber. 2008. Openstreetmap: User-generated street maps. *IEEE Pervasive computing* 7, 4 (2008), 12–18.

[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin* 40, 3 (2017).

[15] Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[17] Philip A Knight. 2008. The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM J. Matrix Anal. Appl.* 30, 1 (2008), 261–275.

[18] Marius Leordeanu and Martial Hebert. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Vol. 2. IEEE, 1482–1489.

[19] Linna Li and Michael F Goodchild. 2011. An optimisation model for linear feature matching in geographical data conflation. *International Journal of Image and Data Fusion* 2, 4 (2011), 309–328.

[20] Yu Liu, Qian Ge, Wei Luo, Qiang Huang, Lei Zou, Haixu Wang, Xin Li, and Chang Liu. 2023. Graphmm: Graph-based vehicular map matching by leveraging trajectory and road correlations. *IEEE Transactions on Knowledge and Data Engineering* 36, 1 (2023), 184–198.

[21] Eliane Maria Loiola, Nair Maria Maia De Abreu, Paulo Oswaldo Boaventura-Netto, Peter Hahn, and Tania Querido. 2007. A survey for the quadratic assignment problem. *European journal of operational research* 176, 2 (2007), 657–690.

[22] Sébastien Mustière and Thomas Devogele. 2008. Matching networks with different levels of detail. *GeoInformatica* 12 (2008), 435–453.

[23] Paul Newson and John Krumm. 2009. Hidden Markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*. 336–343.

[24] Atsumasa Okada, Daisuke Hirouchi, Nobuhisa Matsuda, and Takahiro Miyauchi. 2017. 1:25,000 Urban Active Fault Map "Nakatsugawa".

[25] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. 2018. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks* 108 (2018), 533–543.

[26] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.

[27] Huimin Ren, Sijie Ruan, Yanhua Li, Jie Bao, Chuishi Meng, Ruiyuan Li, and Yu Zheng. 2021. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1410–1419.

[28] Michal Rolínek, Paul Swoboda, Dominik Zietlow, Anselm Paulus, Vít Musil, and Georg Martius. 2020. Deep graph matching via blackbox differentiation of combinatorial solvers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*. Springer, 407–424.

[29] Juan J Ruiz, F Javier Ariza, Manuel A Urena, and Elidia B Blázquez. 2011. Digital map conflation: a review of the process and a proposal for classification. *International Journal of Geographical Information Science* 25, 9 (2011), 1439–1466.

[30] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4938–4947.

[31] Richard Sinkhorn and Paul Knopp. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.* 21, 2 (1967), 343–348.

[32] Abhilshit Soni and Sanjay Boddhu. 2022. Finding map feature correspondences in heterogeneous geospatial datasets. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Geospatial Knowledge Graphs*. 7–16.

[33] Volker Walter and Dieter Fritsch. 1999. Matching spatial data sets: a statistical approach. *International Journal of geographical information science* 13, 5 (1999), 445–473.

[34] Chao Wang, Jing Liu, Kai Wu, and Zhaoyang Wu. 2021. Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection. *IEEE Transactions on Evolutionary Computation* 26, 2 (2021), 304–318.

[35] Runzhong Wang, Junchi Yan, and Xiaokang Yang. 2021. Neural graph matching network: Learning lawler's quadratic assignment problem with extension to hypergraph and multiple-graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 9 (2021), 5261–5279.

[36] Yue Wang and Justin M Solomon. 2019. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*. 3523–3532.

[37] Tonglong Wei, Youfang Lin, Yan Lin, Shengnan Guo, Lan Zhang, and Huaiyu Wan. 2024. Micro-Macro Spatial-Temporal Graph-Based Encoder-Decoder for Map-Constrained Trajectory Recovery. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[38] Christopher E White, David Bernstein, and Alain L Kornhauser. 2000. Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies* 8, 1-6 (2000), 91–108.

[39] Kelvin Wong, Yanlei Gu, and Shunsuke Kamijo. 2020. Mapping for autonomous driving: Opportunities and challenges. *IEEE Intelligent Transportation Systems Magazine* 13, 1 (2020), 91–106.

[40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* (2018).

[41] Chaolong Ying, Yingqi Ruan, Xuemin Chen, Yaomin Wang, and Tianshu Yu. 2025. Neural Graduated Assignment for Maximum Common Edge Subgraphs. *arXiv preprint arXiv:2505.12325* (2025).

[42] Chaolong Ying, Xinjian Zhao, and Tianshu Yu. 2024. Boosting graph pooling with persistent homology. *Advances in Neural Information Processing Systems* 37 (2024), 19087–19113.

[43] Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. 2019. Learning deep graph matching with channel-independent embedding and hungarian attention. In *International conference on learning representations*.

[44] Tianshu Yu, Runzhong Wang, Junchi Yan, and Baoxin Li. 2021. Deep latent graph matching. In *International Conference on Machine Learning*. PMLR, 12187–12197.

[45] Andrei Zanfir and Cristian Sminchisescu. 2018. Deep learning of graph matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2684–2693.

[46] Ayman Zeidan, Eemil Lagerspetz, Kai Zhao, Petteri Nurmi, Sasu Tarkoma, and Huy T Vo. 2020. Geomatch: Efficient large-scale map matching on apache spark. *ACM Transactions on Data Science* 1, 3 (2020), 1–30.