Robust 3D Point Cloud Registration Based on Deep Learning and Optimization

by **Guofeng Mei**

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

Supervisor: Jian Zhang

Co-Supervisor: Qiang Wu

School of Electrical and Data Engineering

Faculty of Engineering and IT

University of Technology Sydney

August 12, 2023

Certificate of Authorship / Originality

I, Guofeng Mei, declare that this thesis is submitted in fulfillment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

List of Publications

Conferences:

1. **Guofeng Mei**, Hao Tang, Liu Juan, Xiaoshui Huang, Jian Zhang, Qiang Wu, and Luc Van Gool, "Unsupervised Probabilistic Approach for Point Cloud Registration," CVPR 2023.

2. Qianliang Wu, Yaqi Shen, Haobo Jiang, **Guofeng Mei**, Yaqing Ding, Lei Luo, Jin Xie, Jian Yang. "Graph Matching Optimization Network for Point Cloud Registration." IROS 2023.

3. **Guofeng Mei**, Fabio Poiesi, Cristiano Saltori, Jian Zhan, and Nicu Sebe. "Overlap-guided Gaussian Mixture Model for Point Cloud Registration," WACV 2023.

4. **Guofeng Mei**, Cristiano Saltori, Fabio Poiesi, Jian Zhang, Elisa Ricci, Nicu Sebe, and Qiang Wu. "Data Augmentation-free Unsupervised Learning for 3D Point Cloud Understanding," BMVC 2022 (Oral).

5. **Guofeng Mei**, Xiaoshui Huang, Jian Zhang, and Qiang Wu. "Overlap-guided Coarse-to-fine Correspondence prediction for Point Cloud Registration," ICME 2022 (Oral).

6. **Guofeng Mei**, Xiaoshui Huang, Jian Zhang, and Qiang Wu. "Partial Point Cloud Registration via Soft Segmentation," ICIP 2022.

7. **Guofeng Mei**, Xiaoshui Huang, Juan Liu, Jian Zhang, and Qiang Wu. "Unsupervised Point Cloud Pre-training via Contrasting and Clustering," ICIP 2022.

8. Xiaoshui Huang, **Guofeng Mei**, and Jian Zhang, "Feature-metric Registration: A Fast Semi-Supervised Approach for Robust Point Cloud Registration without Correspondences." CVPR 2020.

Journals:

9. Xiaoshui Huang, **Guofeng Mei**, and Jian Zhang. "Cross-source point cloud registration: Challenges, progress and prospects." Neurocomputing 2023.

10. Juan Liu, **Guofeng Mei**, Xiaoqun Wu, and Yuanqing Xia. "A unified inferring framework of multiplex epidemic networks under multiple interlayer interaction modes." IEEE TNSE 2023.

11. Youjie Zhou, **Guofeng Mei**, Yiming Wang, Fabio Poiesi, Yi Wan, "Attentive Multimodal Fusion for Optical and Scene Flow." RA-L 2023.

12. Xiaoshui Huang, Yangfu Wang, **Guofeng Mei**, Zongyi Xu, Yucheng Wang, Jian Zhang*, and Mohammed Bennamoun. "Robust Real-world Point Cloud Registration by Inlier Detection," CVIU 2022.

Abstract

3D rigid point cloud registration dedicates to estimating rotation and translation that register point clouds, potentially partially overlapped, into a coherent coordinate system. Registration is an essential but challenging technique in robotics and visual computing. Several efforts have been devoted to developing stable and efficient algorithms to improve registration efficiency and accuracy. Especially learning-based approaches have monopolized recent advances due to the development of point cloud representation learning and differentiable optimization. However, most existing learning-based point cloud matching methods suffer one or more of the following limitations: (1) depending on supervised information from manually labeled data, which is tedious and labor-intensive, (2) suffering from performance degradation to handle point-cloud pairs with large rotation, partial overlaps, and density variations, (3) without integrating point and structure matchings into one stage for searching correspondences, and correspondences are obtained by nearest neighbor search (NN) of local feature descriptors, resulting in high outlier rates, (4) relying on attention mechanisms to simulate soft matching with high computation and memory cost, mainly when being applied to points of a larger number.

This thesis is conducted using optimization theory and deep learning techniques to alleviate the limitations mentioned above. For one thing, it aims to develop unsupervised methods for learning feature representations of point clouds to reduce reliance on human annotations. Another focus of this thesis is on formulating optimization techniques to address registration tasks involving large rotations effectively. Finally, it is also dedicated to designing algorithms to establish more accurate correspondences for point cloud registration with low partial overlaps and density variations. To fulfill these goals, Chapter 3 proposes a soft clustering-based unsupervised algorithm to learn distinctive point cloud representations. The proposed method does not depend on data augmentation, which differs from previous unsupervised works. Chapter 4 extends a correspondence-free method to solve point cloud matching with large rotations and partial overlaps. Expressly, a rotation-based unsupervised method is first provided to learn rotation-sensitive features. Then, a beam search-based scheme incorporates networks to initialize correspondence-free methods to solve large rotation registration. Finally, a clustering-based soft-segmentation approach is employed to solve point cloud alignment with partial overlaps.

Chapter 5 develops a fused optimal transport-based algorithm to establish more correct correspondences in the detected overlapping regions for solving partial overlap registration by integrating point and structure matching. Chapter 6 integrates the overlap scores into a probabilistic registration method to cope with point cloud registration with partial overlaps and density variations. Furthermore, it also provides a clustering-based attention model that simulates matching with low computation and memory cost. All the proposed point cloud matching methods are evaluated on many registration benchmarks, showing their potential to contribute to registration development.

Keywords: Point cloud, registration, large rotation, partial overlaps, point matching, structural matching, unsupervised learning

Acknowledgements

In a mere blink of an eye, three and a half years have passed since I began my Ph.D. studies. During this time, I have witnessed the COVID-19 pandemic firsthand. The sudden and rapid virus spread brought chaos and uncertainty to our daily lives. It was a challenging time for everyone, including myself. Despite the challenges and hardships, it also taught me valuable lessons about resilience, adaptability, and the importance of teamwork in overcoming obstacles.

As I approach the end of my Ph.D. studies, I am filled with gratitude and appreciation for the journey that I have been through. It is with great gratitude and humble appreciation that I express my heartfelt thanks to all who have helped and supported my Ph.D. journey during the pandemic.

First and foremost, I would like to express my deepest gratitude to my principal supervisor, Prof. Jian Zhang, for providing me with invaluable professional guidance, unwavering support, and encouragement throughout my Ph.D. studies. His expertise and wisdom have been instrumental in shaping my research and helping me grow as a scholar. I would also like to express my sincere appreciation to my co-supervisor Prof. Qiang Wu, whose help and guidance during my Ph.D. was crucial for sure no doubt. Further, I would like to express my gratitude to the following collaborators: Professors Nicu Sebe, Elisa Ricci, Mohammed Bennamoun, Luc Van Gool, and Drs. Xiaoshui Huang, Fabio Poiesi, Hao Tang, Litao Yu, and Cristiano Saltori. Their valuable feedback and support have not only helped me revise my manuscript but also inspired me to enhance my research skills and broaden my perspective. I am also thankful to my colleagues and friends at the UTS Multimedia and Data Analytics Lab: Zhibin Li, Yongshun Gong, Lu Zhang, Huaxi Huang, Anan Du, Lingxiang Yao, Wenbo Xu, Mingzhe Wang, Ming Cheng, and all other lab members. Their encouragement and support kept me motivated even during the most challenging times.

Of course, I would like to express my heartfelt gratitude and affection towards my family, who have inspired and supported me throughout my academic journey. Their constant love and encouragement have given me the strength to persevere and achieve my goals.

Finally, I would like to acknowledge UTS for providing financial and material support for my

Ph.D. studies and life in Australia.

This thesis would not have been possible without the support and assistance of so many people, and I am genuinely grateful to all of them.

Guofeng Mei
August 12, 2023
Sydney, Australia

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the prior background knowledge in point cloud registration, research challenges, research significance and goals, as well as research contributions.

1.1 Background

The rapid progress of 3D-capturing sensors, including 3D scanners, LiDARs, and RGB-D cameras, led to more convenient and effective ways to process 3D data, allowing us to understand environments better. 3D data can be a depth image, a point cloud, a mesh, or a computed tomography. In addition, the relatively mature reconstruction algorithms from 2D images make the 3D point cloud accessible. Among all these modalities, this thesis dedicates to point cloud data, one preferred data format representing the objects in the 3D world.

A point cloud comprises a discrete collection of unordered 3D points positioned in a 3D space, depicted by their coordinates on axes and optional attributes like RGB color, normal, and intensity information. Figure 1.1 provides six visual examples of point clouds captured by different approaches, and the greater density of a point cloud means that the point cloud contains more details of objects. The point clouds exhibit inherently unstructured and invariant permutations of their members [1], but it contains much spatially geometrical information. Compared to 2D data, 3D point clouds hold several advantages for machines and humans to understand the surrounding environment better. For example, (i) 3D point cloud provides more useful geometric, shape and scale information than 2D images. (ii) 3D pose of views estimated from the 3D point cloud is more accurate than those recovered from 2D images. As such, these characteristics of point cloud allow it to provide great value in a broad spectrum of applications, ranging from the very large, such as in aero triangulation in smart city reconstruction, to the very small, including particle physics or microbiology. Nevertheless, capturing the whole view range of scans using a single sensor is only sometimes feasible. Therefore, transforming a

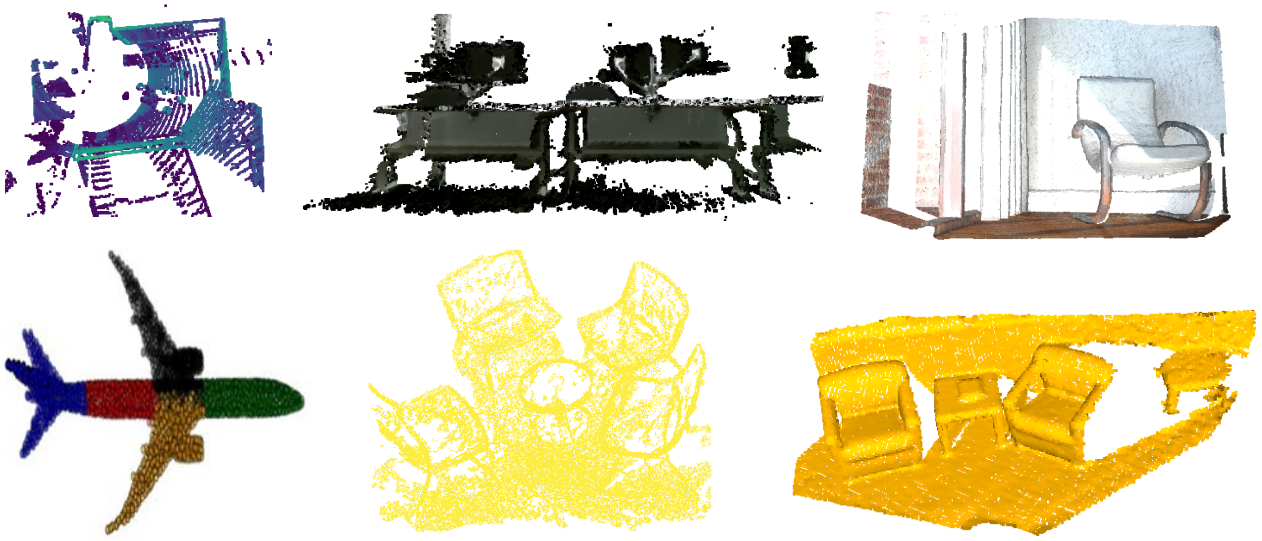


Figure 1.1: Some examples of point clouds captured by different approaches or devices. The greater density of a point cloud, the more detailed the scene representation is captured.

collection of raw point clouds into a large coherent 3D scene is required, usually involved in point cloud registration.

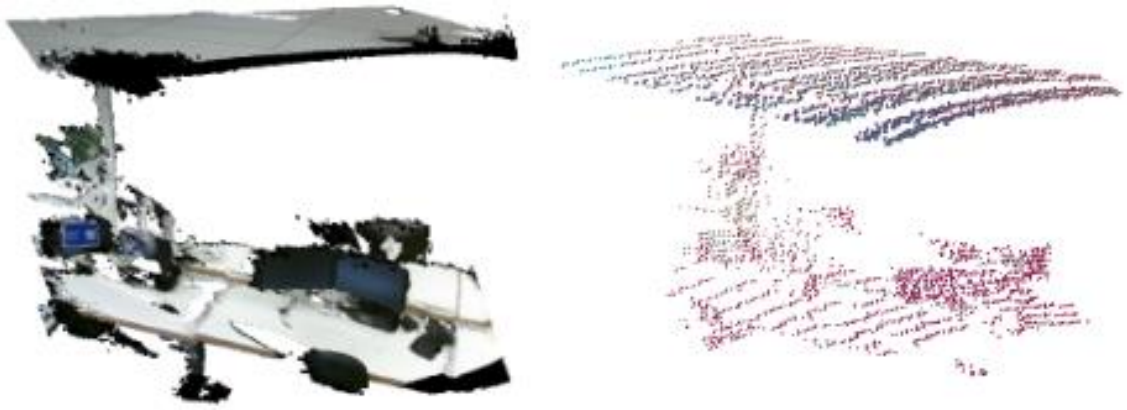
Point cloud registration aims to calculate rotations as well as translations that align multiple point clouds captured from different viewpoints of the same scene, potentially containing different densities, noise, outliers, missing data, and partial overlap, into the specified coordinate system [2]. After being first introduced in the late 1990s, it has been a topic of intense research and exploration for many years [3], and spurred the development of numerous algorithms. According to transformation types, the overall registration can be classified into rigid and nonrigid. The former only involves rigid rotation and translation in the pose, while nonrigid registration deals with the more complex problem of affine transformation, making it a more challenging task. It also can be broadly described as either pairwise or group-wise point cloud registration, with consideration for the number of point cloud sets involved [4]. Pairwise registration involves aligning two point clouds from the same scenes but in different coordinate systems. On the other hand, group-wise registration entails matching point clouds from the same scenes across more than two coordinate systems, and it can also be seen as a series of pairwise alignments. Furthermore, based on the type of features utilized, point cloud registration can be categorized into two groups: non-learning and learning-based methods. Non-learning registration methods typically formulate an iterative optimization process to calculate and determine the motion (rotation and translation). The representatives are the EM-type [5] approaches, such as ICP [6] and its variants [7], [8]. EM-based strategies are developed by minimizing a geometric projection error that alternatively solves two sub-problems: (i) estimating point

correspondences from the input point clouds using estimated transformation and (ii) searching the transformation aligned to the point clouds using the estimated correspondences. However, most EM-type methods require an appropriate initialization and usually converge to the local minima. Different from the non-learning methods, learning-based approaches calculate the geometric transformation through the use of the features learned from the point clouds. These approaches have monopolized current registration advances owing to the advancement of differentiable optimization and the success of deep learning [9]. These approaches, such as PointNetLK [1] and Deep Closet Point (DCP) [10], achieve notable performance since they are more robust and faster than non-learning-based approaches on different datasets. This thesis main focuses on learning-based pairwise registration under a rigid transformation.

Current learning-based registration techniques can be broadly categorized as *correspondence-free* [1], [11] or *correspondence-based* [12]–[14]. The former seeks to minimize the discrepancy between the global feature vectors derived from two given point clouds. These global features are generally obtained by considering the whole points in a point cloud, which makes correspondence-free approaches inadequate to handle real scenes with partial overlaps [3], [12] or large rotation. Correspondence-based methods first extract local features used for establishing point-level [11], [12], [14], [15] or distribution-level [16] correspondences, and finally, estimate the pose from those correspondences. However, point-level registration commonly underperforms when conditions are involved in varying point densities or repetitive patterns [17]. The problem is particularly noticeable in indoor environments where the lack of texture or the repetition of patterns can often dominate the field of view. For example, the feature-matching recall of recent FCGF [18] on 3DMatch [19] dropped from 95% to 80% when the inlier ratio was set to 0.2. That means more than 20% pairs contain more than 80% outliers. Distribution-level registration compensates for the shortcomings of point-level methods and aligns two point clouds without establishing direct point correspondences. Unfortunately, the most current methods are inflexible and cannot effectively handle point clouds with partial overlaps in real scenes [20]. Additionally, the effectiveness of learning-based techniques is heavily reliant on a vast amount of accurate ground truth transformations or matches as guidance for the training of the models. Obtaining the necessary ground truth is often challenging or expensive, which hinders their practical use in the real world [21].

1.2 Research challenges

Point clouds are typically acquired through various time frames, views, or devices, which challenges the registration problem, including large rotation, noise and outliers, density variations, and partial overlaps [22]. For instance, Kinect and LiDAR differ in the number of points they capture in a specific area, resulting in varying densities. This can be seen in Figure 1.2, which



3D sensor (Kinect) **3D sensor (Lidar)**

Figure 1.2: Density variations attributed to the point clouds are caught from Kinect and Lidar, respectively.

displays point clouds with different densities. LIDAR performs optimally in all types of lighting. However, LIDAR experiences difficulties in severe weather conditions, including snow, fog, rain, as well as air-borne dust particles. As a result, this leads to issues with missing data in certain areas, causing partial overlaps. Figure 1.3 shows the point clouds with different overlap ratios. The smaller overlap ratios result in more registration challenges. Additionally, point clouds of large-scale scenes can have millions of points and require efficient algorithms to handle the computational complexity of registration. To sum up, different data problems lead to various registration difficulties. The section will introduce the challenges in detail. The mainstream registration approaches of point clouds comprise three modules: feature extraction, correspondence prediction, and pose estimation [23]. So, the challenges are introduced according to each module. Finally, this section also presents the challenges of correspondence-free methods in this section.

Overlap rate is 0.1 **Overlap rate is 0.3** **Overlap rate is 0.5**

Figure 1.3: Point clouds with different overlap ratios. The smaller overlap ratios result in more registration challenges. Image is from [14].

1.2.1 Challenges on feature extraction

Learning discriminative and robust point cloud features is crucial in learning-based registration [24]. However, the effectiveness of these techniques is contingent mainly upon having abundant ground truth, such as transformations or correspondences, to serve as the supervision signal for model training [25]. Needless to say, the required ground truth is typically difficult or costly to acquire, thus hampering their application in the real world [21]. To handle ground-truth labeling issues, great efforts [2], [11], [21], [26] have been devoted to unsupervised deep point cloud registration. The existing methods mainly lie in auto-encoders [11], [21], [26] with a reconstruction loss or contrastive learning [18], [27], [28] with data augmentation. Although significant progress has been achieved, there are still certain obstacles that need to be addressed. Firstly, some methods depend on the point-level loss, such as Chamfer distance in auto-encoder [26], finding it challenging to handle large-scale scenarios due to computational complexity. Secondly, many pipelines [2] apply fixed/handcrafted data augmentation to generate transformations or correspondences, leading to sub-optimal learning. This is because they can only fully use the cross information of partially overlapping point clouds with geometric labels, and the fixed augmentation neglects the shape complexity of the samples [29]. Second, the accuracy of the alignment decreases significantly when used on new 3D scans that have significant rotational changes because the features that have been learned are sensitive to rotation [30].

1.2.2 Challenges on correspondence prediction

Pointwise feature matching is often used to establish matches of two input point clouds [31]. The central concept behind pointwise matching is that a pair of points of two point clouds having the most similar feature representations are identified as the corresponding points. However, the putative correspondences produced by the pointwise feature similarity contain many false matches and outliers [23]. This issue is especially prominent in indoor environments, where the presence of low-texture areas or plain patterns often dominates the field of view. Many recent works show that the following four factors will cause false matches and outliers: first of all, some inliers are assigned to outliers because the inliers and outliers are equally treated in the correspondence prediction stage [23]; second, some points of point clouds should be ignored as they can not find a match due to partial overlap and density variations [24]; thirdly, the scale of the correspondence search space increases quadratically with the number of points of two given point clouds; finally, more than pointwise matching is needed to determine the correct correspondences by reason of the ambiguous and repeated patterns in the 3D point clouds [15]. However, eliminating the outliers and obtaining accurate correspondences are the remaining challenges in the 3D point cloud alignment. It has been stated that more than relying

solely on point-point feature similarity is required to estimate accurate correspondences in 3D acquisition point clouds, as there are often instances of ambiguous and plain patterns present. An alternative to finding point-to-point correspondences is using distribution-to-distribution matching through probabilistic models [25]. These probabilistic registration techniques showed greater robustness to noise and density variations than their point-to-point counterpart [32]. However, they usually do not distinguish the points in the overlap and non-overlap regions and typically require their inputs to share the same distribution parameters (e.g., Gaussian Mixture Models [33]). Therefore, these algorithms are limited to either registering two complete point clouds or one complete and one partial point cloud. In real-world scenarios, partial-to-partial registration is frequently necessary, but distribution matching-based methods can suffer from suboptimal performance because of the differences in distribution parameters.

1.2.3 Challenges on pose estimation

Kabsch algorithm [34] is widely applied to calculate the transformation from a set of correspondences and associated weights. The estimated correspondences in this field are often contaminated by outliers caused by partial overlapping and other factors. In order to filter out false matches, RANSAC [35] is commonly used. However, RANSAC is not suitable for training pipelines as it is non-differentiable. This has led to the development of direct regression of transformation parameters has emerged as a new trend.

1.2.4 Challenges on correspondence-free methods

Correspondence-free approaches minimize the discrepancy between the global feature vectors derived from two given point clouds [36]. These global features are typically computed based on all the points of a point cloud, making correspondence-free approaches inadequate to handle real scenes with low partial overlap [3], [12]. This is because it is unreasonable to expect a neural network to capture the same features from two given point clouds that have significant differences. Besides, these methods exhibit inadequate performance when applied to point clouds that contain multiple objects, such as indoor environments.

1.2.5 Summary

The success of deep learning has led to significant advancements in point cloud alignment techniques [17]. These techniques have accomplished high precision and speed by integrating deep learning with traditional optimization. However, several limitations still impede the effectiveness of practical applications. By way of illustration, training a feature extractor typically requires supervised information from manually labeled data, which is laborious and time-consuming. For another, most existing registration approaches are locally optimal and tend to

fail when the relative rotation is large or low over partial. Therefore, it would be beneficial to develop registration algorithms to alleviate the dependence on labeled data and are suitable for large rotations and low overlaps.

1.3 Research significance and goals

The significance of point cloud registration is evident due to its indispensable function in various applications [22]. Specifically, in scenarios where multiple point clouds are captured from different sources, it is possible to encounter situations where certain regions lack data or have missing information. Therefore, registration techniques are required to complete the point cloud missing areas. There is also an application in large-scale aircraft measurement. The multi-station scanning strategy is commonly adopted to scan the entire aircraft surface since it is sometimes impractical to capture the whole surface of an aircraft at a single location. Therefore, a critical component of aircraft detection is transforming point clouds collected from different coordinate systems into a global coordinate system. Further, the efficiency of the registration process in real-time is essential for the navigation of robots and autonomous driving vehicles [24]. Besides, the utilization of various sensors has proven to be beneficial in constructing more precise and extensive 3D models. LiDAR, in particular, is well-known for its accuracy in creating 3D models, but increasing its resolution can be pretty expensive. Hence, a more cost-effective solution is to combine different sensors and create a fused solution that generates accurate and economical 3D point clouds. To achieve this, cross-source point cloud registration becomes an essential tool [24]. This is because this technique involves registering point clouds obtained from different sources, ensuring the accuracy of the 3D model and providing a comprehensive representation of the object or scene [37].

In summary, developing registration algorithms to support applications such as robotic navigation [38], autonomous driving [39], and augmented and virtual reality [40] would be advantageous as these applications require 3D registration as a critical component in their operation. Additionally, point cloud registration can enable change detection and monitoring over time and facilitate the analysis and interpretation of 3D data.

The primary objectives of this thesis involve the development of optimization and deep learning methodologies that address the challenges associated with point cloud registration. To be more detailed, the main goals are:

- To build some unsupervised methods to reduce the dependence on labeled data;
- To extend the learning-based local registration methods to solve large rotation cases;
- To devise some methods to establish more accurate correspondences for registration,

especially for point clouds with low partial overlaps;

- To enable the distribution-based registration methods applied to partial overlap situations;

1.4 Research contributions

This thesis aims to develop algorithms based on deep learning and optimization to approximate global solutions. To overcome previously discussed challenges and issues, this thesis has devised corresponding solutions. The study contributions are presented as follows:

- `Contribution 1`: To avoid the inconvenience of building chains of ad-hoc combinations of data augmentations for unsupervised learning, this thesis develops an augmentation-free unsupervised approach to extracting informative point-level representations of 3D point clouds without data augmentation. The algorithm can be utilized in any network designed for extracting point-wise features. It is independent of large batch sizes and negative samples (Chapter 3); This thesis also presents a rotation prediction-based approach to extract rotation-awareness features for correspondence-free registration methods without depending on any labeled data. The proposed method has the ability to detect rotational changes, making it ideal for correspondence-free registration techniques (Chapter 4);

- `Contribution 2`: To address the challenge of registering large rotations utilizing correspondence free local techniques, this thesis presents a hierarchical and greedy algorithm that employs a beam search scheme. This algorithm offers a significant improvement in solving large rotation registration problems. An unsupervised soft-segmentation algorithm based on soft-clustering provided in Chapter 3 is also developed to achieve partial shape alignment. This study is the pioneer in utilizing beam search to forecast transformations for deep point cloud registration that involves large rotations (Chapter 4);

- `Contribution 3`: To establish more accurate point correspondences for partial overlap registration, this thesis presents a joint model that combines overlap scores and pointwise and structural matchings. The model adopts a coarse-to-fine hierarchical structure based on fused optimal transport. The pointwise matching-based correspondence estimation is transformed into an optimization problem based on Wasserstein distance. On the other hand, structural matching is expressed as a Gromov-Wasserstein distance-based optimization problem that predicts matches. The structural difference is calculated in both Euclidean and feature spaces. The model also includes an overlap detection module to learn point-wise features and overlap scores. Ultimately, the correspondences can be predicted by solving a fused Gromov-Wasserstein objective (Chapter 5);

- `Contribution 4`: To overcome the limitations of pointwise correspondence-based methods in noise and density variations, this thesis presents a probabilistic registration approach, which calculates the optimal transformation from matched Gaussian Mixture Model (GMM) components. The registration problem is reformulated as aligning two Gaussian mixtures by minimizing a distribution discrepancy measure between them. Furthermore, a Transformer-based detection module is also introduced to identify overlapping regions. By doing this, the GMM representations of the input point clouds are under the guidance of overlap scores computed by the proposed detection module. Additionally, a cluster-based loss is introduced to ensure that the network learns a consistent GMM representation across both feature and geometric spaces rather than fitting a GMM in a single feature space (Chapter 6).

1.5 Thesis organization

This thesis is organized as follows:

Chapter 2 provides a comprehensive overview of the definition and methods of point cloud registration. The non-learning-based registration techniques and current learning-based methods are discussed in detail.

Chapter 3 proposes a soft-clustering-based unsupervised approach to learning informative point-level representations of 3D point clouds without data augmentation. The proposed method is also applied to softly segment a point cloud into parts for Chapter 4 to solve partial overlap registration in an unsupervised way.

Chapter 4 builds an unsupervised method to train feature extractors without labeled data and extends correspondence-free methods to adapting large rotation and partial overlapping registration based on Chapter 3.

Chapter 5 provides a correspondence prediction method based on fused optimal transport to estimate correspondences in the detected overlap regions.

Chapter 6 presents a new overlap-guided probabilistic registration method that calculates the optimal transformation based on matched Gaussian Mixture Model (GMM) components.

Chapter 2

Literature Survey

This chapter reviews several related research in 3D point cloud registration, starting with conventional non-learning methods and proceeding to the latest deep learning techniques. Additionally, this chapter reviews works related to optimal transport, which have been extensively used in this thesis. Lastly, this chapter analyzes the datasets utilized in point cloud registration as well as the widely used data augmentation techniques on 3D point clouds.

2.1 Non-learning-based Point Cloud Registration

Non-learning-based registration adopts optimization algorithms to recover the transformation. These methods have a long history of research and development. This review will examine different types of optimization-based methods based on their optimization strategies, such as EM [5]-based, RANSAC [41]-based, graph-based, and so forth.

2.1.1 EM-based registration

The prominent EM-based algorithm for pairwise registration is Iterative Closest Point (ICP) [42], [43]. It iteratively alternates between estimating the transformation and searching for correspondences [10], [37] by solving an L_2-optimization problem. ICP is favored for its simplicity and relatively fast processing time, especially when implemented with kd-trees for efficient closest-point searching. However, vanilla ICP can converge to spurious local minima due to the non-convexity of the objective function, requiring appropriate initialization. Additionally, ICP faces challenges arising from differences in point densities, presence of noise, outliers (unexpected points), occlusions (missing points), partial overlaps, and limited one-to-one correspondences between two point clouds [24].

To overcome these limitations of ICP, researchers have proposed various variants. One such

variant is the Levenberg-Marquardt ICP [44], which employs the Levenberg-Marquardt algorithm [44], replacing singular value decomposition with gradient descent and Gauss-Newton [45] approaches. This accelerates data registration convergence while maintaining high accuracy. Point-to-plane and plane-to-plane criteria have also been developed to aid in constructing correspondences and reduce the algorithm's sensitivity to noise. The point-to-plane criterion calculates the distance between a point in one point cloud and the plane defined by the nearest neighbors in the other point cloud. The goal is to minimize this distance for all corresponding points. The plane-to-plane criterion, on the other hand, calculates the distance between two planes defined by their normal vectors and a point on each plane. The goal is to find the transformation that minimizes the distance between the two planes, which is a more global approach than the point-to-plane criterion. Generalized ICP [8] modifies the standard ICP algorithm by representing each point as a Gaussian distribution during the optimization process, which leads to a Maximum Likelihood Estimation (MLE) based loss function. Moreover, it uses KD-trees to establish discrete correspondences based on the raw points, making the registration process more robust in noise and outliers. In this modified algorithm, the optimization objective is a weighted sum of point-to-plane and point-to-point distances, where the covariance matrices of the corresponding points determine the weights.

While these methods are valuable for local registration, Go-ICP [46] stands out as a global approach that efficiently solves the point cloud alignment problem using the Branch-and-Bound (BnB) optimization [47] framework, eliminating the need for prior information about correspondences or transformations. The branches in BnB represent different choices for matching points between the two point clouds. The algorithm evaluates the cost of each node based on the sum of squared distances between the matched points and prunes branches that cannot lead to a better solution than the best one found so far. Go-ICP has shown superior performance over existing ICP algorithms on various benchmark datasets, especially for point clouds with substantial noise or outliers and large rotations. However, the computational complexity of Go-ICP makes it less practical for real-time applications. To address this limitation, several extensions and variants of Go-ICP have been proposed, including parallelization, acceleration techniques, and hybrid methods that combine BnB with other optimization approaches. These efforts aim to make Go-ICP more suitable for real-time applications while preserving its global registration capabilities. Non-linear ICP [48] leverages the smooth and differentiable properties of the Huber loss function [49] to efficiently reduce the influence of outliers. The non-linear aspect of the algorithm comes from the fact that it uses a non-linear optimization technique to find the optimal transformation parameters, which enables it to handle more complex transformations than traditional linear ICP methods.

Nonetheless, real-time processing is a challenge for ICP-based methods due to their slow speed,

and the models may get stuck in local minima if the initial estimate is not accurate. Additionally, these methods may not be suitable for point clouds with partial overlap or contaminated by outliers.

2.1.2 RANSAC-based registration

RANSAC-like randomized estimation [35], [50] is another widely used family of techniques for robust finding the correct correspondences for registration. Initially, several control pairs are picked randomly from two given point clouds, which are then used to compute the transformation matrix. Secondly, the estimated motion is applied to estimate the correspondences and counts the number of inliers (consensus set), and the transformation matrix is recalculated. Lastly, an iterative process is carried out to locate the most extensive consensus set, considered the final solution. Compared with EM-based methods, it lies in its robustness against outliers and its ability to handle challenging registration scenarios, such as low overlapping registration.

Many variants have been developed to enhance the specific elements of RANSAC, with the aim of making the original algorithm more efficient. For example, MLESAC [51] and MAGSAC++ [52] generalize RANSAC by adopting more advanced scoring functions such as likelihood and re-weighted least-squares. Unlike traditional methods like RANSAC, which rely on random sampling, MLESAC uses a maximum likelihood approach to iteratively refine a model until it fits the data optimally. This allows MLESAC to handle outliers and noise more effectively than RANSAC, which tends to struggle when dealing with complex data sets with a large number of outliers. MAGSAC improved MAGSAC++ by performing a two-step approach to model fitting. The first step is the robust estimation of the initial model parameters using MAGSAC, which is a geometric algorithm that is able to handle large amounts of outliers. In the second step, MLESAC is used to refine the model parameters, which is a maximum likelihood algorithm that is able to improve the accuracy of the initial estimates. GroupSAC [53] utilizes a hierarchical sampling paradigm based on the assumption that inliers have more remarkable similarities among themselves. Locally optimized methods, including LORANSAC [54] and GC-RANSAC [55] perform the RANSAC step once the best model has been discovered so far. The LORANSAC algorithm works by randomly selecting a subset of data points and fitting a model to them. The model is then evaluated on the remaining data points, and the inliers (points that fit the model) are used to update the model. This process is repeated until a satisfactory model is found or a maximum number of iterations is reached. GC-RANSAC uses a combination of graphical models and belief propagation to improve the accuracy and efficiency of the model fitting process. It also includes a built-in outlier rejection mechanism, which helps to identify and remove any erroneous data points that may negatively impact the quality of the model. One-point RANSAC [56] aims to reduce the iterations of RANSAC

by estimating scale and translation parameters separately. It begins by randomly selecting a minimal set of points (in this case, a single 3D point) and estimating the camera pose using a closed-form solution. Next, the algorithm evaluates the fit of the estimated model by counting the number of inliers and outliers. If the number of inliers exceeds a certain threshold, the algorithm terminates and returns the estimated camera pose. Otherwise, the algorithm repeats the sampling and estimation process until a satisfactory result is obtained. DSAC [57] is the first work that provides a differentiable RANSAC-like estimator using a soft probabilistic hypothesis selection. Neural-guided RANSAC [58] uses the inlier probabilities provided by neural networks to guide the hypothesis sampling. It utilizes both neural networks and RANSAC to boost object detection and segmentation accuracy and speed. The neural network is then applied to predict the shape and location of objects, while RANSAC is used to refine the objective parameters based on the predicted location.

While RANSAC-like algorithms are generally efficient, their time complexity grows exponentially with an increasing proportion of outliers [59]. Additionally, they may not always provide the optimal solution. Despite these drawbacks, RANSAC remains a powerful tool for robust point cloud registration tasks, particularly in the presence of outliers and challenging datasets.

2.1.3 Graph matching-based registration

The core concept behind utilizing graph matching (GM) for registration is to represent point clouds as graphs. Graph matching is the process of determining the correspondences between nodes of two graphs, typically composed of nodes and edges. This task involves utilizing information from both nodes and edges to formulate an optimization problem. According to the constraints of objective functions, GM methods can be coarsely classified into first-order, second-order and high-order approaches. The former searches for correspondences by only comparing the similarity nodes in a graph [60], involved in solving a linear assignment problem. The comparison of the similarity of nodes and edges is referred to as the second-order GM method. In contrast, the high-order GM method involves comparing more than two nodes, wherein the similarity of triangle pairs is also considered [61].

Second-order optimization falls under quadratic assignment problems (QAP) [62], which is burdensome to be solved since it is NP-hard. Numerous approximation algorithms have thus been developed to tackle the QAP, resulting in various approaches that can be utilized to obtain a locally optimal solution for graph matching. One widely explored approximation algorithms are relaxation based, which can fall into three categories: double stochastic, spectral, and semi-definite programming relaxations. The double stochastic relaxation involves transforming the optimization of the GM into solving non-convex QAP, which is achieved by substituting the integrity constraints with corresponding box constraints. Spectral relaxation is primarily

focused on relaxing the permutation constraints inherent in graph matching, formulated through the utilization of spectral features of adjacency matrices. The core concept behind semi-definite relaxation involves linearizing quadratic terms by introducing new variables and adding a convex semi-definite constraint to establish a connection between the original variables and the new variables. Further, many improved variants of relaxation have also been explored. For example, Leordeanu *et al.* [63] proposed a spectral graph matching method that uses spectral relaxation to approximate the QAP problem by solving a semi-definite programming (SDP) relaxation to relax the non-convex constraint using a convex semi-definite. For instance, Almohamad *et al.* [64] applied linear programming techniques to approximate the quadratic cost function. Zhou *et al.* [65] devised a factorized graph matching approach, which factorizes a large pairwise affinity matrix into smaller matrices and utilizes a path-following optimization algorithm to solve the GM problem.

High-order graph matching algorithms are advantageous as they are invariant to affine variations, such as scale differences. For example, Zass *et al.* [66] designed a probabilistic approach to solve the high-order graph matching problem. Recently, Zhu and colleagues [67] developed the elastic net method, which incorporates a flexible net constraint into the tensor-based graph matching model to control the trade-off between sparsity and accuracy of the matching results. All these methods are affine-invariant and have been applied in a variety of applications such as image registration, object recognition, and machine learning. In the case of point cloud registration, the systematic graph matching (CSGM) method by Huang *et al.* [68] employs a linear program to find correspondences by solving a second-order graph-matching problem. High-order GM, proposed in [61], involves using a tensor power iteration algorithm to optimize the relaxed form of high-order GM in the integer domain. The resulting solution is then projected onto the feasible solution space.

In summary, graph matching-based methods for point cloud registration offer unique advantages compared to RANSAC-based and ICP-based methods. These approaches leverage the inherent structural information present in point clouds and represent them as graphs, where nodes correspond to points and edges encode the relationships between points. By formulating registration as a graph matching problem, these methods can exploit geometric constraints and global contextual information, leading to more accurate and robust registration results, especially in scenarios with partial overlaps and complex transformations.

2.1.4 Probability-based registration

Unlike traditional optimization-based approaches such as RANSAC, ICP, and graph matching, probability-based registration algorithms can provide not only correspondences but also a measure of confidence for each correspondence. It models the point clouds as probability dis-

tributions, often via the use of GMMs, and performs registration either by solving correlation-based or EM-based optimization pipelines [69], [70]. The correlation-based methods [16], [70] first build GMM probability distributions for two given point clouds that need to be matched. After that, the transformation is determined by minimizing the difference between the two distributions as measured by a specific metric or divergence. For example, GMMReg [70] assumes that the data can be represented by a mixture of Gaussian distributions, and estimates the parameters of the mixture model using the expectation-maximization (EM) algorithm. The resulting model is then used to perform robust regression by minimizing a weighted sum of squared errors, where the weights are determined by the likelihoods of the data points under the mixture model. One of the key advantages of GMMReg is its ability to handle outliers and non-Gaussian noise. The algorithm can identify and ignore outliers by assigning them low weights in the regression objective function. This results in a complex optimization problem that involves nonconvex constraints that are not guaranteed to have a unique solution [71].

Different from correlation-based methods, the EM-based approaches, such as Coherent Point Drift (CPD) [72], JRMPC [73], and FilterReg [74], represent the geometry of one point cloud using a GMM distribution over 3D Euclidean space. The transformation is then calculated by fitting another point cloud to the GMM distribution utilizing the maximum likelihood estimation (MLE) pipeline. The key feature of CPD is its ability to handle non-rigid deformations between point sets. It can also handle missing data and outliers in the point sets. JRMPC is based on joint registration and mutual principal component analysis, and it is able to handle multi-modal images or point clouds.

In sum, the probability-based methods are robust to noise, outliers and density variation [16]. Most of them utilize robust discrepancies to mitigate the influence of outliers by greedily aligning the largest possible fraction of points while being tolerant of a small number of outliers. However, if outliers dominate, the greedy behavior of these methods easily emphasizes outliers, leading to degraded registration results [73]. Furthermore, the parameters of the distribution are not ensured to be uniform across different views [16].

2.1.5 Other registration methods

There are several other conventional techniques used for registration that have also achieved remarkable outcomes. For instance, FGR [75] takes a different approach from RANSAC-like methods by optimizing a Geman-McClure [76] cost-based correspondence objective function using a graduated nonconvex strategy, resulting in outstanding performance. Meanwhile, TEASER [77] redefines the registration problem as an intractable optimization and provides verifiable conditions to determine whether the output solution is optimal. This method remains effective even when dealing with extremely high outlier rates. Additionally, Chen *et al.* employed a

novel two-stage strategy that focuses on maximizing the inlier set, as discussed in their publication [78].

2.2 Deep learning-based point cloud registration

The reliance on handcrafted features for distinguishing correspondences heavily depends on the expertise of designers, leading to limited generalizability and robustness across various applications. To address these limitations, efforts have been made to develop and employ deep learning algorithms for point cloud registration. This section examines key components in learning-based point cloud registration pipelines, which have been instrumental in recent advancements in registration techniques. It commences with an exploration of deep learning models on 3D data, as extracting features from the point cloud is a crucial aspect of learning-based approaches. Lastly, the section delves into learning-based point cloud registration approaches, categorized based on whether they utilize learned features for searching correspondences. These methods are further divided into correspondence-free and correspondence-based approaches.

2.2.1 Deep learning on 3D point cloud

Deep learning has successfully solved a range of 3D point cloud problems. PointNet [79] and DeepSets [80] are the pioneering architectures that directly process unordered and unstructured 3D points by independently extracting the features for each element in the point cloud and combining them using invariant permutation operations. Though efficient, PointNet and DeepSets only encode global representation aggregated from the pointwise features and fail to capture local structures, impeding application to tasks involving local geometry [81].

The improved version of PointNet, PointNet++ [82], adopts sampling and grouping operations to hierarchically extract features from point patches. PointNeXt [83] further improved PointNet++ by introducing an inverted residual bottleneck design and separable MLPs. Similarly, PointCNN [84], PointConv [85], and Relation-Shape CNN [86] also focus on extracting more semantic features from the local region by separating points into scales or bins, and then aggregating these features by concatenation [87]. RPM-Net [88] has the ability to infer both movable parts and their corresponding motions from a single 3D point cloud shape, which may be un-segmented and incomplete. This is achieved through an encoder-decoder pair with LSTM components that predict a sequence of pointwise displacements for the input shape. As a result, the network is able to learn the movable parts and segment the shape based on its motion. By recursively applying RPM-Net to the segmented parts, the network can further predict more detailed motions and achieve a hierarchical object segmentation. DGCNN [89] handles point clouds by utilizing a disordered graph neural network with dynamic graph construction that

allows the model to capture the local and global structures of point clouds. This is achieved by first constructing a k-nearest neighbor graph for each point in the input point cloud, then using a graph convolution operation to update each point's features based on its neighbors' features. The graph is dynamically updated during training, allowing the model to adapt to different structures and patterns in the point clouds. To improve DGCNN performance, PAConv [90] uses a plug-and-play convolutional operation for deep representation learning on 3D point clouds. It incorporates a learnable weighting function to adjust the filter's receptive field according to the input's local geometry. This allows the network to handle position, orientation, and scale variations. AdaptConv [91] strives to establish the relationship by adaptively learning features from point patches in a hierarchical manner and capturing the different relationships between points from various semantic parts accurately. EC-Net [92] presents an edge-conscious method to boost the merging of point cloud sets. This approach devises a regression model capable of retrieving not only the 3D point coordinates but also the distances from the points to edges using upsampled features. An edge-sensitive joint loss function is also utilized to minimize the distances from the output points to both the surface and the edges. KPConv [93] adopts discrete kernel points to imitate a continuous convolution kernel. The core idea behind KPConv is to replace the standard convolutional kernel used in traditional CNNs with a set of learnable kernel points that adapt to the local geometry of the input point cloud. Each kernel point is associated with a weight and a radius, which defines the neighborhood of points that contribute to the output of the KPConv layer. Sun *et al.* [94] proposed a point-to-surface representation for 3D point cloud feature extraction considering both the point and the geometric surface simultaneously. Ding *et al.* [95] developed a perturbation learning-based point cloud upsampling method to generate uniform, clean, and dense point clouds. LGA [96] aimed to a framework that aggregates geometries in a layer-by-layer manner to achieve lossless compression of LiDAR point cloud geometry.

Transformer attention [97] has recently achieved great success in point cloud tasks, as it is invariant to the permutation of input tokens and can learn long-range dependencies. Applying Transformer attention to 3D point clouds [98], [99] is thus natural since point clouds are collections of permutation-invariant points in 3D space [98]. Figure 2.1 depicts a diagram of a Transformer model that employs dot-product attention. The model takes a feature map $\boldsymbol{\mathcal{F}}$ as input and uses three MLPs to generate feature maps $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ from $\boldsymbol{\mathcal{F}}$. The dot-product of $\boldsymbol{Q}$ and $\boldsymbol{K}$ is used to calculate attention weights, which are then applied to $\boldsymbol{V}$ to produce a weighted feature map. The weighted feature map is added with the raw feature map, and an MLP is applied to produce the final feature map $\boldsymbol{\mathcal{F}}^T$. The pioneering work Point Transformer [98] applies self-attention to local neighborhoods around each point and the encoding of positional information in the network. It achieved exceptional results in point cloud classification and segmentation assignments. PATs [100] introduces a Self-Attention structure to capture

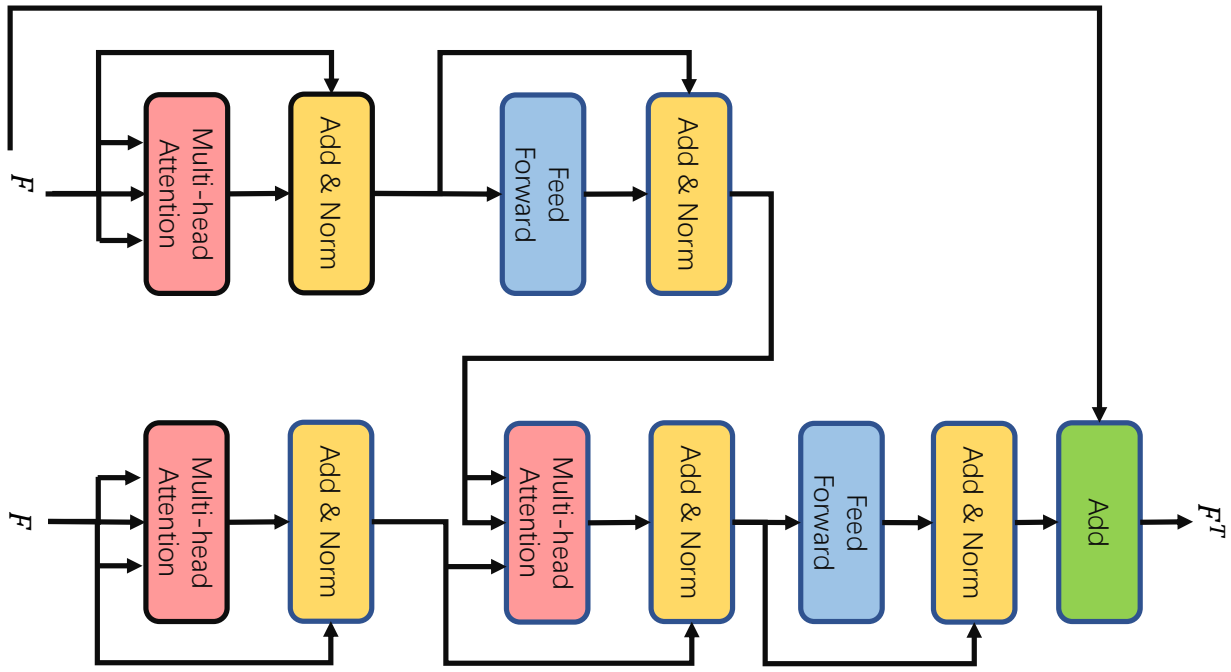


Figure 2.1: The diagram of Transformer based on dot-product attention. It enhances the feature representation of the point cloud by selectively attending to important points, which improves the discriminative power of the feature extractor.

the good relations between points. Transformer3D-Det [101] proposed to tackle the challenge of 3D object detection by utilizing the attention mechanism to capture the relationships between neighboring clusters, thereby generating more precise voting centers. 3DCTN [102] provided a hierarchical structure with graph convolution to reduce the computational and memory costs of traditional attention. Geometric Transformer [9] is a tool for acquiring knowledge about geometric features in order to facilitate reliable feature matching. This is accomplished by encoding both pairwise distances and triplet-wise angles, rendering the approach resilient in low-overlap situations while also invariant to rigid transformations. AWT-Net [103] first generates point-wise wavelet coefficients, which split each point into high or low sub-band components, followed by Transformer to fuse features from different but integrated sub-bands.

Despite their impressive achievements, these methods necessitate supervised information for feature learning. This reliance on annotated data hinders the integration of point cloud models into new environments with limited labeled information. Thus, it becomes crucial to devise techniques that lessen the need for annotated samples while maintaining satisfactory performance in point cloud understanding tasks through deep learning. In this regard, the adoption of an unsupervised learning approach shows promise in reducing the dependence on labeled data. Current 3D sensing modalities have enabled the generation of extensive unlabeled 3D point cloud data [104]. Therefore, recent efforts [21], [105]–[107] have been dedicated to exploring methods for training a network by making full use of the unlabeled data. This has boosted a

recent line of works on learning discriminative representations of 3D objects using unsupervised approaches [104], [108], [109].

Unsupervised point cloud representation learning approaches can be classified into generative and discriminative methods. The former applies self-reconstruction [87] or adversarial learning to jointly learn representation and model point clouds. For example, FoldingNet [110] leverages a graph-based encoder and a folding-based decoder to deform a standard 2D grid into the surface of a point cloud. FoldingNet is based on the concept of folding a 2D sheet of paper to create a 3D object. The model takes a 2D image of an object as input and generates a 3D representation of the object by folding the image along its creases. L2G Auto-encoder [111] employs a local-to-global autoencoder to capture the local and global information of point clouds simultaneously. It further uses a loss function that combines the reconstruction error and the group sparsity constraint. The group sparsity constraint promotes the learning of feature representations that are not only sparse but also structured in a way that is meaningful for the given task. In [112], a graph-based decoder with a learnable graph topology is used to push the codeword to preserve representative features, which can help improve the quality of the reconstructed data. In [113], a combination of hierarchical Bayesian and generative models are trained to generate plausible point clouds. GraphTER [114] self-trains a feature encoder by reconstructing node-level transformations from the representations of both the original and altered graphs. However, generative models are sensitive to transformations, weakening the learning of robust point cloud representations for different downstream tasks. Moreover, it is not always feasible to reconstruct the shape from pose-invariant feature representations [115].

Contrastingly, discriminative methods are based on auxiliary handcrafted prediction tasks to learn point cloud representations. For instances, Jigsaw3D [116] uses a 3D jigsaw puzzle approach as the self-supervised learning task. The key idea behind Jigsaw3D is to break down a 3D shape into smaller, manageable pieces and then train a neural network to reconstruct the original shape from these pieces, which is similar to solving a jigsaw puzzle. Recently, contrastive approaches [27], [81], [115], [117], which are robust to transformation, achieved sophisticated performance. Info3D [115] maximizes the mutual information between the 3D shape and its altered version resulting from geometric transformation. PointContrast [27] is the first to explore a unified paradigm of contrastive learning for self-training 3D point cloud representation, which is achieved by maximizing the similarity between positive pairs of points while minimizing the similarity between negative pairs. Many works show that the effectiveness of contrastive methods is contingent upon the correct design of negative mining strategies and the right choice of data augmentations that should not affect the semantics of the raw point clouds. Transformer-based networks have a range of unsupervised methods available to them. Point-BERT [118] is an example that partitions a point cloud into several local point

patches, uses a discrete Variational AutoEncoder Point Cloud Tokenizer to generate discrete point tokens that capture important local details, and then randomly masks and processes a few input point cloud patches through the backbone Transformers. The primary objective of pre-training is to recover the original point tokens at the masked positions, with guidance from the point tokens obtained from the Tokenizer.

Some works also concentrate on the unsupervised registration of point clouds. There are some noteworthy works. PPF-FoldNet [119] utilizes a UNet-like encoding scheme and a FoldingNet-like decoder on 4D-PPFs to learn transformation-invariant features. This framework is built on the point-pair feature (PPF) descriptor and uses a folding-based neural network to perform recognition and pose estimation. The PPF descriptor captures the geometric properties of a pair of points in a point cloud, such as their relative positions and surface normals. 3DFeat-Net [120] proposes a weakly supervised that leverages alignment and attention mechanisms to acquire feature correspondences from 3D point clouds tagged with GPS/INS information without requiring explicit specification. The network inputs a set of triplets consisting of anchor, positive, and negative point sets. The self-training of the model utilizes the triplet loss function, which seeks to decrease the dissimilarity between the anchor and positive counterparts while increasing the dissimilarity between the anchor and negative point clouds. In a similar fashion, SiamesePointNet [121] employs a hierarchical encoder-decoder architecture to generate descriptors for points of interest. This architecture is trained to map points that are both geometrically and semantically similar to each other, resulting in their proximity in descriptor space. PRNet [2] designs a detector for keypoints and uses the correspondences between keypoints to register point clouds that only partially overlap, doing so in a self-supervised manner based on data augmentation. PRNet fails to converge to good results when trained on 3DMatch. In [105], cycle consistency is employed as a pretext task to self-train the feature extractor. However, the cycle-consistency loss may perform poorly in cases of partial overlap as outliers may need to form a closed loop effectively. RIENet [21] provided a method to identify the inlier according to the graph-structure difference between the neighborhoods in an unsupervised manner.

Although encouraging results have been achieved, some challenges remain to be addressed. Firstly, they depend on the point-level loss, such as Chamfer distance in auto-encoder [26], finding it challenging to handle large-scale scenarios due to computational complexity. Secondly, many pipelines [2] apply fixed/handcrafted data augmentation to generate transformations or correspondences, leading to sub-optimal learning. This is because they can only fully use the cross information of partially overlapping point clouds with geometric labels and the lack of consideration of the shape complexity of the samples in the fixed augmentation [29].

The upcoming discussion will explore the application of deep learning to address point cloud registration, which can be divided into correspondence-free and correspondence-based methods.

2.2.2 Correspondences-free approaches

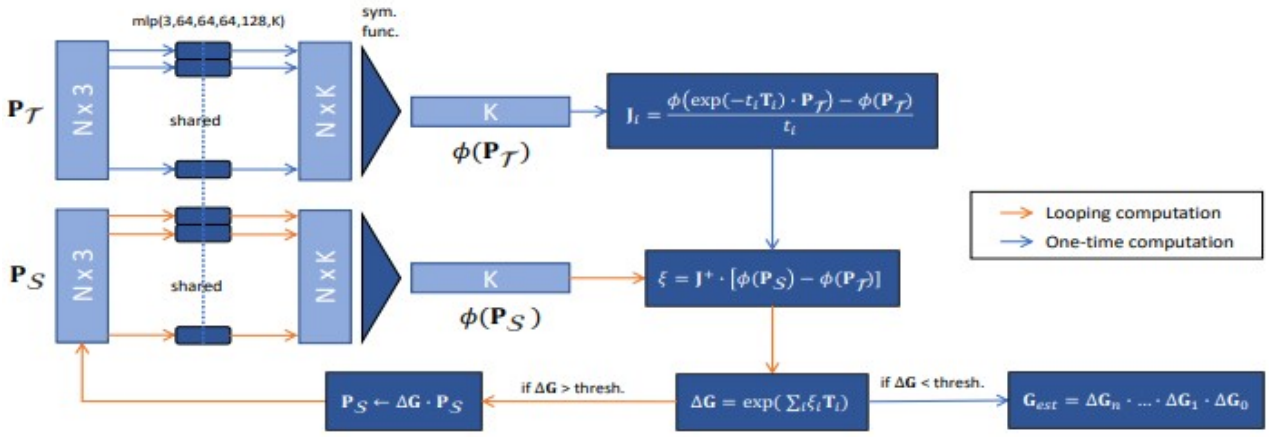


Figure 2.2: The framework of the PointNetLK. It utilizes the efficient inverse compositional Lucas-Kanade algorithm (IC-LK) to estimate the skew-symmetric matrix representation iteratively by minimizing the misalignment between two point clouds features produced by PointNet. Image is from [1], Fig. 2.

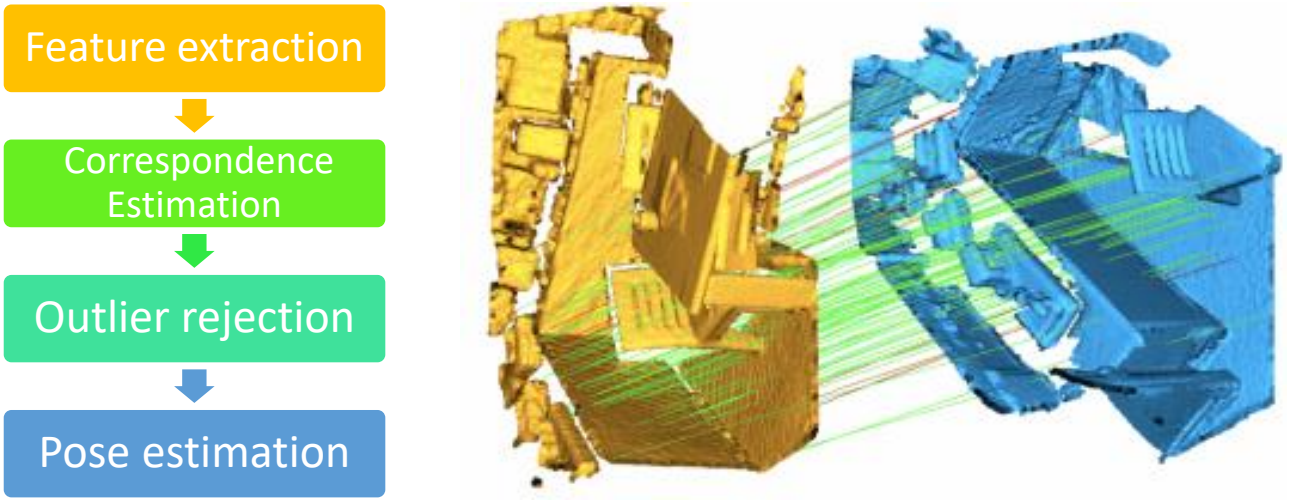
The core idea of correspondences-free registration approaches [1], [11], [17], [36] is to determine the transformation by minimizing the dissimilarity between the global features extracted from two input point clouds that can be registered [3]. This method requires the extracted global representation to be sensitive to rotation and translation. PointNetLK [1] is a representative method of correspondences-free approaches, which utilizes the efficient inverse compositional Lucas-Kanade algorithm (IC-LK) to estimate the skew-symmetric matrix representation iteratively by minimizing the misalignment between two point clouds features produced by PointNet. Note that a finite difference gradient algorithm is applied to approximate Jacobian for IC-LK. PointNetLK is a milestone work that translates the registration problem from the previous point-to-point matching to minimize the global feature difference of two point clouds. Figure 2.2 shows the framework of the PointNetLK. Feature-metric registration (FMR) [11] inherits the advantages of PointNetLK and further improves PointNetLK with an autoencoder and a chamfer distance loss. FMR utilizes an autoencoder to extract the global features more impressionable to the pose since the decoder module enforces the encoder to keep pose information. Meantime, a semi-supervised or unsupervised approach can be adopted to train this model, thus reducing the dependency on label data. Similarly to PointNetLK, PCRNet [122] first leverages PointNet to extract global features, which are then applied to regress transformation. Different from PointNetLK, for the pose estimation module, the extracted global features are concatenated and provided as input of an MLP network to regress the transformation parameters. Like IC-LK, PCRNet introduced the iteration strategy to improve accuracy and obtain robust performance. PCRNet displays improved generalizability compared to Point-

NetLK. However, it is not as resilient to noise. OMNet [123] provides an iterative end-to-end method for learning masks that can reject regions without any overlap. The technique is developed using a convolutional neural network (CNN) that can extract and learn the feature maps from the partial point clouds. EquivReg [124] employs a neural network with SO(3)-equivariance to convert rotation search to global feature matching. The network architecture is designed to be invariant to rotations in three dimensions, ensuring that the output remains consistent regardless of the input's orientation. This allows EquivReg to effectively search for correspondences between features in different views without explicitly computing rotations. Instead, EquivReg relies on global feature matching, which involves comparing sets of features across different views to identify similar patterns. UPCR [125] to make correspondence-free methods fit for point cloud registration with partial overlaps and outliers based on a representation separation perspective. The key idea behind the representation separation perspective is to extract multiple features from the point clouds that capture different aspects of the underlying structure. These features are then combined in a non-linear way to compute the registration transformation without needing correspondence. FINet [126] adopts a dual branches structure by separating the features into rotation and translation components, recognizing their distinct solution spaces. Additionally, the feature extractor is augmented with interactive modules to facilitate data association. Furthermore, FINet incorporates a transformation sensitivity loss to enhance the features' attentiveness to rotation and translation.

These correspondence-free techniques are immune to density variations and noise and do not necessitate the exploration of correspondences at the point or distribution level. Furthermore, these methods depend on the derivation of all-encompassing representations to decrease the point cloud's dimensions, ensuring that the algorithm's time complexity remains constant as the number of points increases. Nevertheless, they heavily rely on sufficient overlaps between two point clouds and suffer from performance degradation for only partially overlapped point clouds. To summarize, this kind of methods that do not require correspondence offer the benefits of speed, high precision, and resilience to noise and density fluctuations when the point clouds overlap entirely. Still, they need help to handle partial-to-partial or large rotation point cloud alignment effectively. Their performance strongly relies on the features, and their generalization capability needs improvement and applicability to the real scene.

2.2.3 Correspondences-based approaches

The fundamental concept behind correspondences-based methods involves obtaining deep features on a per-point or per-patch basis in order to accurately determine correspondences. Then, the correspondences are utilized to estimate the transformation using optimization algorithms, such as SVD and IC-LK. These algorithms are a significant part of deep point cloud registration



(a) Process **(b) Point correspondences**

Figure 2.3: Correspondence-based point cloud registration. (a) Process of correspondence-based registration; (b) Extracted point correspondences based on feature similarity.

methods [3], typically consisting of four key components: feature extraction, correspondence search, outlier elimination, and pose estimation.

Feature extraction. Correspondence-based registration commonly incorporates 3D extractors to capture localized geometric descriptors for the purpose of feature matching. These extractors can be categorized into two main groups: patch-based and fully convolution-based. The input of the patch-based extractors is the local patch in the point cloud. Patch-based extractors function by taking local patches within the point cloud as input. Noteworthy among these approaches is 3DMatch [31], which stands as a pioneering method. It leverages a Siamese 3D Convolutional Neural Network (CNN) to produce feature representations tailored to local regions. Specifically, it trains the model by consuming inputs of volumetric 3D patches and outputs a 512-dimensional feature vector for each patch, which is served as the feature representation of the local regions. To address the efficiency and rotation sensitivity issues of 3DMatch, Gojcic *et al.* [127] propose a pre-processing method to transform the 3D local patches canonical representations based on a local reference frame (LRF), followed by extracting the per point local feature descriptors from local canonical representations using a network. To be more precise, a Local Reference Frame (LRF) is constructed by performing an eigendecomposition on the covariance matrix of all the points. Once each patch of point clouds is aligned to its associated LRF, a Gaussian smoothing technique is employed to obtain a Smooth Density Value (SDV) voxelization of the input grids. Subsequently, this SDV is passed through 3DSmoothNet for feature extraction. PPFNet [119] extracts local descriptors based solely on geometry while also possessing a strong awareness of the global context. This method first extracts rotation-invariant patch descriptions based on PPF [128] from each point cloud patch. Then, these descriptions are fed into a PointNet [79] to extract a local feature, followed by a max-pooling

operation to extract a global feature. The MLP block takes both the global and local features as input and produces the ultimate descriptor for correspondence search. PPF-FolderNet [26] further adopts an autoencoder module to alleviate the dependence on manual annotation of matching point clusters. The optimization of the network involves the utilization of Chamfer loss, which calculates the disparity between the input and output. Deng *et al.* utilized PPF-FoldNet and PC-FoldNet auto-encoders in their work [129] to extract both pose-invariant and pose-variant local features. The invariant features were utilized to recognize matching key points, while the pose-variant local descriptors associated with the matched key points were combined and input into the RelativeNet, which produced predictions for the relative pose.

To learn rotation invariant features, SpinNet [30] is focused on developing a low-dimensional embedding for point clouds within a feature space that possesses both rotational and translational invariance. It utilizes a novel cylindrical representation to transform the input 3D local patch and then applies advanced neural layers to learn informative and comprehensive local patterns effectively. The point cloud is first presented as a series of planes that are oriented in order to give an approximation of the object's surface. The orientation of these planes is established through the local geometry of the surface, while the distances between the planes and points provide information regarding the shape of the point cloud. Despite the fact that SpinNet produces accurate registration with excellent generalization, the construction of local features in a patch-wise manner is quite time-consuming, which limits its practicality. YOHO [130] strives to extract rotation-equivariant local features built on the icosahedral group features. It employ both rotation-invariance and rotation-equivariance to identify correspondences between point clouds and estimate plausible rotations. Rather than relying on an external LRF, YOHO achieves rotation invariance through neural network-based feature extraction on the $SO(3)$ group, which leverages the robustness of neural networks to variations in point cloud density and noise. Furthermore, by utilizing rotation-equivariance, YOHO is able to estimate rotations with just one matched point pair.

Although these approaches demonstrated impressive results, they have a few limitations. One of which is that patch data requires a large amount of GPU memory, which can be a bottleneck in terms of computational resources.

On the other hand, fully convolutional techniques generate dense features for the entire point cloud in a single forward pass and implement a contrastive loss function on individual points, rather than patches. These methods not only achieve the leading performance levels but also maintain low inference times. FCGF [18] introduces a new type of feature descriptor, called geometric features, which encodes both the geometric and spatial information of the points in the input cloud. These features are extracted using a network that is trained to predict the local geometry of each point in the voxel grid. However, the learned point local features are

still sensitive to large rotations. KPConv [93] adopts kernel-based point convolution or submanifold sparse convolution to extract 3D geometric features from point clouds for geometric correspondence. Yang *et al.* [131] proposed introduced a novel approach to merging high-level and low-level local geometric features by leveraging a neural network model optimized within the triplet framework. This model combines local geometric features in a non-linear fashion in Euclidean spaces. To train the model, the researchers utilized an enhanced triplet loss that utilizes all pairwise relationships within the triplet. The resulting fused descriptors are equivalent in performance to deeply learned descriptors derived from raw point clouds, yet are more lightweight and rotation-invariant. RPSRNet [132] adopts a unique approach to represent point clouds using a 2^D-tree and hierarchical deep feature embedding in the neural network. And the network employs an iterative transformation refinement module to enhance the feature-matching accuracy in intermediate stages. The main contribution is that it adopts a unique approach to represent input point clouds using a 2^D-tree and hierarchical deep feature embedding in the neural network. Additionally, the network employs an iterative transformation refinement module to enhance the feature-matching accuracy in intermediate stages.

In general, the full convolution-based approaches exhibit significantly greater speed compared to their patch-based counterparts. On the other hand, patch-based descriptor methods excel at capturing intricate local features and demonstrating strong generalization capabilities. Regrettably, both these techniques are susceptible to noise and lack effective strategies for handling scenes with substantial noise interference.

Correspondence Estimation. Predicting reliable correspondences is the key to registration success in the correspondence-based method. Most methods estimate correspondences by calculating the similarity of learned or hand-crafted geometric features. The pioneering approach proposed by DCP [10] utilizes a dynamic graph convolutional neural network to extract features and an attention module to produce soft matching pairs. It borrows ideas from the classic ICP pipeline while striving to avoid the associated sub-optimal issue. RPMNet [133], and REGTR [134] perform feature matching by integrating the Sinkhorn algorithm or Transformer [97] into a network to generate soft correspondences from local features. One of the key advantages of RPM-Net is its robustness to noise and occlusion. This is achieved through the use of a "soft assignment" approach, which allows the algorithm to match points even if they are not exact matches. Su *et al.* [135] employed the Wasserstein distance [136] to deal with the 3D shape matching and surface registration problem. FIRE-Net [137] analyzes the interaction between source and target point clouds from different levels. It uses a Combined Feature Encoder to extract interactive features within each point cloud and enhance the network's ability to describe local geometry. It also employs a Local Interaction Unit and Global Interaction Unit to facilitate interaction between point pairs across two point clouds, allowing for increased

awareness between similar point features and the global perception of each other's features. VRNet [138] initially creates corresponding virtual points (VCPs) by employing a soft matching matrix estimation to calculate a weighted average of the target points. A correction-walk module is then implemented to learn an offset that rectifies VCPs into RCPs, allowing greater distribution flexibility. Lastly, a hybrid loss function is designed to ensure that the learned RCPs conform to the shape and geometric structure of the source while providing ample supervision. DIT [139] employs a combination of techniques to effectively model global relationships and extract structural information from point clouds. These techniques include a Point Cloud Structure Extractor with Transformer encoders, a deep-narrow Point Feature Transformer for deep information interaction across two point clouds, and a Geometric Matching-based Correspondence Confidence Evaluation (GMCCE) method for measuring spatial consistency and estimating inlier confidence. Transformers are used to establish comprehensive associations and directly learn the relative position between points through positional encoding. The triangulated descriptor enables the GMCCE method to evaluate correspondence confidence. S2H [140] adopts a two-step approach for learning a partial permutation matching matrix that avoids assigning corresponding points to outliers and ensures unambiguous assignments. The first step involves solving the soft matching matrix, and the second step projects this soft matrix to the partial permutation matrix through a hard assignment. To accomplish this, the profit matrix is augmented before the hard assignment to obtain an augmented permutation matrix, which is then cropped to achieve the final partial permutation matrix. To ensure end-to-end learning, the learned partial permutation matrix is supervised, but the gradient is propagated to the soft matrix.

Different from soft-matching based methods, DeepICP [141] utilizes different types of deep neural network architectures to create a fully trainable network. The system trains the keypoint detector through this end-to-end architecture, allowing it to ignore dynamic objects and focus on highly distinguishable features of stationary objects, which leads to exceptional robustness. Instead of searching for corresponding points among pre-existing points, the system generates matching points based on learned probabilities among a selection of potential candidates, resulting in improved accuracy during registration. IDAM [142] utilizes a hybrid approach that combines geometric and distance features during its iterative matching process. The methodology involves generating a similarity score for point matching by concatenating the features of the two points of interest. By using this method, the limitations of using inner products to determine pointwise similarity can be overcome. DeepGMR [16] is a probabilistic method that employs a neural network for learning GMM distributions, which are served to search correspondences between points and distribution components. These correspondences are then utilized in the GMM optimization module to calculate the transformation matrix. StickyPillars [143] employs graph neural networks and conducts context aggregation on 3D keypoints

that are sparsely distributed, leveraging a transformer-based multi-head self and cross-attention mechanism. The resulting network output serves as the cost factor for resolving an optimal transport problem, whose solution provides the ultimate matching probabilities. HRegNet [144], specifically designed for large-scale outdoor LiDAR point cloud data, employs a hierarchical approach to extract keypoints and descriptors, resulting in accurate and robust registration. This approach combines features from deeper layers to enhance reliability and shallower layers to provide precise position information. In addition, HRegNet uses bilateral and neighborhood consensus to match keypoints and introduces new similarity features to enhance the correspondence network. Dang *et al.* [145] applied Wasserstein distances to handle 3D point cloud registration. RobOT [146] applied the advanced optimal transport (OT) [136] tools to solve feature matching for large-scale point clouds with more than 10k points. DeepUME [147] utilizes a coordinate system that is invariant under SO(3) to learn a joint resampling strategy and SO(3)-invariant features for point clouds. These features are then used by the geometric UME method for estimating transformations. To overcome the ambiguity that arises in the registration of symmetric shapes in noisy scenarios, the parameters of DeepUME are optimized using a specifically designed metric. UTOPIC [148] concentrates on solving the challenging issue of ambiguous overlap prediction by utilizing a feature extractor to extract shape knowledge through a completion decoder. Furthermore, it generates a geometric relation embedding, allowing Transformer to obtain feature representations that are both transformation-invariant and geometry-aware. PCAM [149] incorporates the multiplication of cross-attention matrices at each point, allowing for the fusion of low-level geometric characteristics and high-level contextual information to accurately identify corresponding points. Additionally, the cross-attention matrices enable the transfer of contextual knowledge between the point clouds at every level, resulting in the generation of high-quality matching features in areas of overlap between the two clouds.

Keypoint-free methods [9], [150] first downsample each point cloud into super-points and then register them by checking whether their neighborhoods (patches) overlap. OCFNet [23] introduced overlap scores to reject non-overlapping regions, consequently guiding the model to match overlapping points with high probabilities. However, most existing methods only consider the point-point similarity while omitting the structural similarity. The structural similarity is also significant in finding accurate correspondences.

Outlier rejection. Even with the quick advancements in 3D local features, the correspondences obtained from feature matching are still vulnerable to outliers, especially when there is a limited overlap of scene fragments [13], [151]. As a result, outlier rejection is essential for enhancing registration precision. The most widely used outlier removal methods are the RANdom SAmple Consensus(RANSAC) [41] and its variants. Branch and bound-based optimization [59]

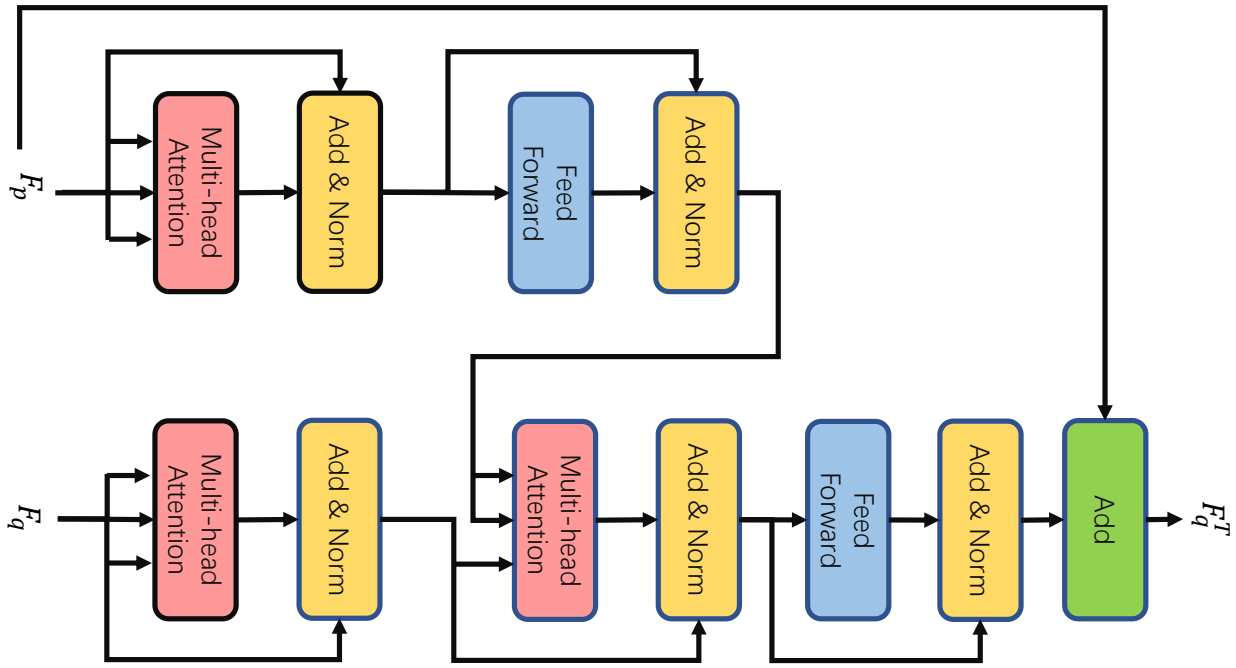


Figure 2.4: The diagram of cross-attention. It takes F_p and F_q as inputs. It allows information from different sources to be selectively attended to and integrated into a single representation.

is also applied to remove the outliers.

Recently, deep learning has emerged as a promising approach for detecting outliers. Pioneer works in this domain include regression-based methods. For instance, DGR [12] employs a six-dimensional convolutional network to predict the likelihood of points being inliers and applies a weighted Procrustes module for transformation calculation. Similarly, 3DRegNet [152] utilizes an inlier prediction model to estimate the probability of points being inliers. RPM [15] focuses on the information exchange between two point clouds to extract distinctive features for each point, enhancing point matching.

Another widely applied strategy is based on spatial compatibility. PointDSC [13] effectively rejects outlier correspondences by incorporating spatial consistency. It includes two key components: a nonlocal feature aggregation module that employs feature and spatial coherence weighting to effectively embed input correspondences, and a differentiable spectral matching module that estimates the confidence of each correspondence with respect to inliers based on the embedded features. SC^2-PCR [153] introduces a second-order spatial compatibility measure to determine the similarity of correspondences, which takes into account global compatibility. In the initial stages, SC^2-PCR yields clustering between inliers and outliers, which is more distinguishable instead of solely focusing on local consistency. The registration pipeline uses a global spectral technique based on this measure to identify reliable seeds. Then, a two-stage strategy is implemented to expand each seed to a consensus set, utilizing the SC^2 measure matrix. TriVoC [154] breaks down the task of identifying the smallest 3-point sets into three sequential

layers. Each layer includes an efficient mechanism for voting and categorizing correspondences, which is based on the constraint that they must be of equal length when compared pairwise. This enables the selection of 3-point sets from reduced correspondence sets independently, following a sorted sequence. As a result, computational expenses are considerably decreased, while the likelihood of obtaining the largest consensus set (as the ultimate inlier set) is substantially increased, provided that a probabilistic stopping criterion is satisfied. DHVR [155] generates a collection of triplets of matched point pairs that are used to perform votes on the 6D Hough space. These point pairs are represented in sparse tensors, followed by employing a fully convolutional refinement network to improve the accuracy of the votes. After picking matched point pairs in the Hough space, a consensus is reached and utilized to make predictions about the final transformation parameters. PointCLM [156] leverages contrastive technique to evenly acquire distributed feature representations of possible correspondences. According to these representations, PointCLM employs an outlier removal approach and a clustering method to effectively eliminate outliers and allocate the reserving correspondences to their corresponding categories. MultiReg [157] clusters the related points into distinct groups based on a distance invariance matrix. This matrix is then generated by verifying the distance coherence between every set of correspondences following the point associations.

Additionally, the use of Transformer architecture has proven useful in detecting overlap regions and, in turn, removing outliers. For instance, Predator [14] integrates a cross-attention module into FCGF/KPConv feature extractor to extract more distinctive features and identify overlap regions. This enhances the extraction of distinctive features and improves correspondence accuracy. The cross-attention mechanism selectively attends to important points in the point cloud, thereby enhancing the feature representation and discriminative power of the feature extractor. Figure 2.4 shows the diagram of cross-attention, which enhances the feature representation of the point cloud by selectively attending to important points, which improves the discriminative power of the feature extractor. PRNet [2] focuses on detecting points in the overlap region and utilizing their features to generate matches. DetarNet [151] utilizes a consensus encoding network to produce a descriptor for each correspondence and an attention block to identify inlier correspondences. To begin, the method introduces a Progressive and Coherent Feature Drift technique to align source and target points in high-dimensional feature space and accurately estimate the resulting translation. Next, a Consensus Encoding Unit generates more unique features for a given set of putative correspondences. Finally, a Spatial and Channel Attention block is employed to construct a classification network that can detect optimal correspondences.

In conclusion, the application of deep learning and innovative techniques like spatial compatibility and Transformer architecture has led to notable advancements in outlier rejection for 3D point cloud registration. These methods play a crucial role in enhancing registration precision

by effectively removing outliers and improving correspondence accuracy.

Transformation estimation. A commonly used method to convert or approximate poses is through solving a least-squares problem utilizing either singular value decomposition (SVD) or the Kabsch algorithm [34], with the option of applying weights. Since both the SVD and Kabsch algorithms can be differentiated, they are appropriate for integrating into a neural network for complete training. Another method involves initially parameterizing the rotation matrix using Lie algebra, then using iterative techniques like Gauss-Newton minimization to calculate the transformation. This section presents a brief summary of the weighted Kabsch algorithm, which is extensively utilized in this thesis. The algorithm begins by computing the centroid, or the geometric center, of both coordinate sets and subsequently aligns the sets by translating them to the centroid. Following this, the algorithm computes the covariance matrix between the sets and employs singular value decomposition (SVD) to identify the optimal rotation matrix that minimizes the root-mean-square deviation. To be more detailed, when given matching score $\mathbf{\Gamma}$, point cloud $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i\}$ and $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_i\}$, it first selects correspondences with confidence higher than a threshold of $\tau > 0$, and further enforces the mutual nearest neighbor (MNN) criteria, which filters possible outliers. The correspondences are defined as:

$$\mathcal{M} = \{(\boldsymbol{p}_i, \boldsymbol{q}_j) \big| \forall (i,j) \in \text{MNN}(\boldsymbol{S}), \boldsymbol{\mu}_{p_i} \geq \tau, \boldsymbol{\mu}_{q_j} \geq \tau\}. \tag{2.1}$$

The next step is to employ the weighted SVD on the correspondences with weights Γ to estimate the rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$. Specifically, a matrix $\widehat{\Gamma}$ is first constructed by selecting the i-th row and j-th column from $\mathbf{\Gamma}$ for each $(\boldsymbol{p}_i, \boldsymbol{q}_j) \in \mathcal{M}$. The weight vector $\boldsymbol{w} = [w_1, w_2, \cdots, w_{|\mathcal{M}|}]$ is denoted for convenience, where $w_i = \sum_j \widehat{\Gamma}_{ij}$. Furthermore, the normalized weight vector is defined as $\bar{\boldsymbol{w}} = [\bar{w}_1, \bar{w}2, \cdots, \bar{w}|\mathcal{M}|] = \frac{\boldsymbol{w}}{|\boldsymbol{w}|_1}$. The transformation T (rotation R and translation $\boldsymbol{t}$) can be calculated by minimizing

$$\sum_{(\boldsymbol{p}_i, \boldsymbol{q}_j) \in \mathcal{M}} \bar{w}_i \|\boldsymbol{p}_i - R\boldsymbol{q}_j - \boldsymbol{t}\|_2^2. \tag{2.2}$$

It can be solved by SVD, more details can be found in DGR [12].

Summary. To sum up, correspondence-based registration of point clouds provides precise alignment and remains a popular, flexible, and time-efficient technique. It offers higher precision compared to correspondence-free methods, making it advantageous for applications requiring accuracy. Correspondence-based registration is versatile, capable of handling various point cloud types, including 3D models, laser scan data, and images. However, it does have some limitations. It is sensitive to noise, large rotations, and partial overlap, often requiring feature matching, which can introduce ambiguities in the registration process. Moreover, the correspondence-based approach may consume considerable time during computation. Despite

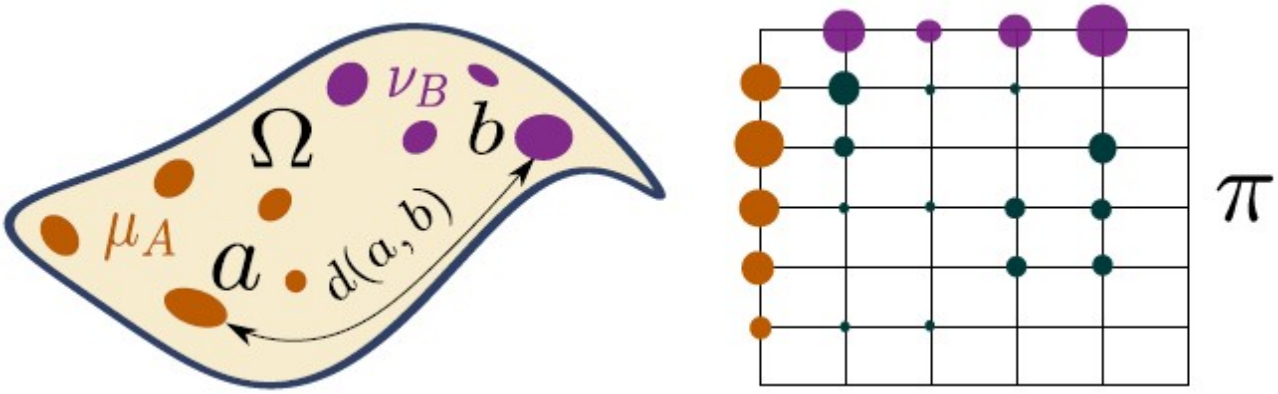


Figure 2.5: The diagram of Wasserstein distance between two discrete measures μ_A and μ_B on a space Ω. Ω is armed with distance metric d. On the left, there are two discrete measures defined on the space Ω, while on the right, there is one admissible coupling π between the two measures. This image is taken from Fig. 5 of [160].

these challenges, correspondence-based registration remains a valuable and widely-used method in the field of point cloud alignment.

2.3 Optimal transport

This section reviews optimal transport (OT) [136] as it has been extensively used in this thesis. It is a method to exploit the best assignment between two general objects, which is widely applied in many machine learning applications [158]. The Wasserstein distance (WD) [159] is a type of distance metric associated with OT, used to measure the difference between probability measures. WD aligns the points by comparing their difference in distributions based on their masses and transportation costs. As shown in Fig. 2.5, WD provides a way to predict the correspondences and measure the similarity between two distributions. However, WD is unsuitable for probability measures lying in different spaces. To bridge this gap, the Gromov-Wasserstein distance (GWD) [136] extends WD to handle the probability distribution distributions whose supports lie on different metric spaces by comparing the similarity between pairwise distances, as shown in Fig. 2.6. The GWD has also been used for shape analysis [161], graph matching [162], etc. The GWD is hard to solve since it is a quadratic programming problem. Moreover, GWD solely concentrates on illustrating the connections between their own components, depicting the structure of the object, yet overlooking feature information. To overcome this limitation, the Fused-Gromov Wasserstein (FGW) distance was proposed in [163] as a combination of Gromov Wasserstein and Wasserstein distances, thus simultaneously taking into account feature and structural information. As fused Gromov-Wasserstein distance compares both GWD and WD, its distance is also a quadratic programming problem. These OT-based distances have

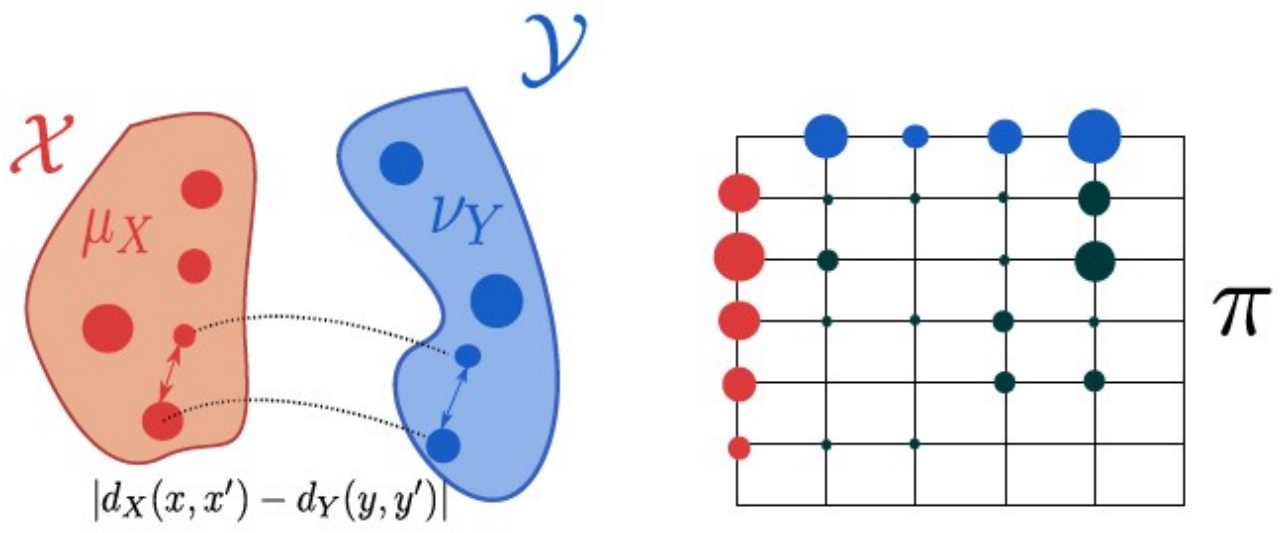


Figure 2.6: The Gromov-Wasserstein coupling is a method for comparing two metric spaces $\mathcal{X} = (\boldsymbol{X}, d_X, \mu_x)$ and $\mathcal{Y} = (\boldsymbol{Y}, d_Y, \mu_Y)$ that have Borel sets belonging to $\sigma-$algebra $\boldsymbol{X}$ and $\boldsymbol{Y}$, with corresponding measures μ_x and ν_Y, and distances d_X and d_Y, respectively. In the left figure, the two metric spaces have nothing in common. In the right figure, a possible coupling is shown. The image is taken from [160], Figure 6.

been widely recognized for their exceptional performance in the fields of machine learning and computer vision, including graph matching [162], shape matching [146], wgan [164], domain adaptation [165], and reconstruction [166]. Nevertheless, OT-based approaches usually suffer the following limitations:

- computational complexity is expensive, limiting the application of optimal transportation to large-scale data analysis.
- finding the GWD distance remains challenging, as it is involved in solving an NP-hard nonconvex quadratic program.
- the GWD is not suitable for partially overlapped shape matching as it can only be applied to metric measure spaces with a probability distribution for comparison.

As a result, recent advances in optimal transport have been monopolized by designing algorithms to reduce the computational complexity of optimal transport ranging from entropic regularization [167], Sliced Wasserstein [168] to Sampled Gromov Wasserstein [169]. Entropic regularization optimal transport is based on the idea of "regularizing" the transport problem by adding a small amount of randomness or uncertainty to the cost function. This randomness is measured using the concept of entropy, which captures the degree of disorder or unpredictability in a system. By introducing entropy into the cost function, entropic regularization optimal transport is able to smooth out the optimization problem and make it more tractable. The

Sliced Wasserstein distance specifically uses a technique called slicing, which involves projecting the distributions onto a lower-dimensional subspace and calculating the Wasserstein distance between the projected distributions. This can make the calculation more efficient and computationally tractable in high-dimensional settings. The Sampled Gromov-Wasserstein algorithm works by randomly sampling subsets of the two datasets and then computing the Gromov-Wasserstein distance between the sampled subsets. This approach makes the computation much faster than the traditional Gromov-Wasserstein algorithm, which requires a brute-force comparison of all data points. To deal with arbitrary positive measures, i.e., the measures are not constrained in the distribution space, Thibault *et al.* [170] applied quadratic divergences and bi-convex relaxation to solve unbalanced or partial Fused Gromov-Wasserstein distance. Optimal transport is also being applied to point cloud registration. However, most of the OT-based approaches utilized dustbins to reject outliers, which can not deal with point clouds with partial overlaps, effectively.

2.4 Data augmentation

The learning-based registration benefits greatly from the use of data augmentations. The stochastic data augmentation module is responsible for randomly transforming any given point cloud, generating two correlated views, considered positive pairs. Various techniques can be used to augment point cloud data, such as jittering, scaling, rotating, flipping, adding noise, and so forth. The following section will list some commonly used data augmentation methods.

- `Random Crop` selects a half-space at random with a direction from $\mathcal{S}^2$ for a given point cloud and moves it in such a way that a proportional amount of points are preserved.
- `Random Rotation` rotates the entire point cloud along a random axis and angle to simulate different viewpoints.
- `Random Jittering`: Random Jittering adds random noise to the point cloud data to simulate real-world variations and make the dataset more robust to noise and other sources of error.
- `Random Translation` randomly translates the entire point cloud in 3D space to simulate different camera positions. This can be used to generate more training data for object detection or segmentation tasks.
- `Random Scaling` scales the entire point cloud in all three dimensions by a random factor to change the size of the point cloud. This can simulate different distances from the object or test algorithms' performance on different scales.

- `Random Sampling` involves reducing the number of points in a point cloud by sampling the data at a lower resolution. This can help improve model efficiency, reduce computational requirements, or generate more training data with varying densities.
- `CutMix` [171] cuts and pastes random patches of two or more point cloud inputs together to create a new, combined point cloud. By utilizing this technique, the quality and diversity of training data can be enhanced, ultimately leading to better performance of machine learning models that utilize point cloud data. Figure 2.7 demonstrates the visualization of mixed samples between a chair and a plane with varying replacement ratios (λ).

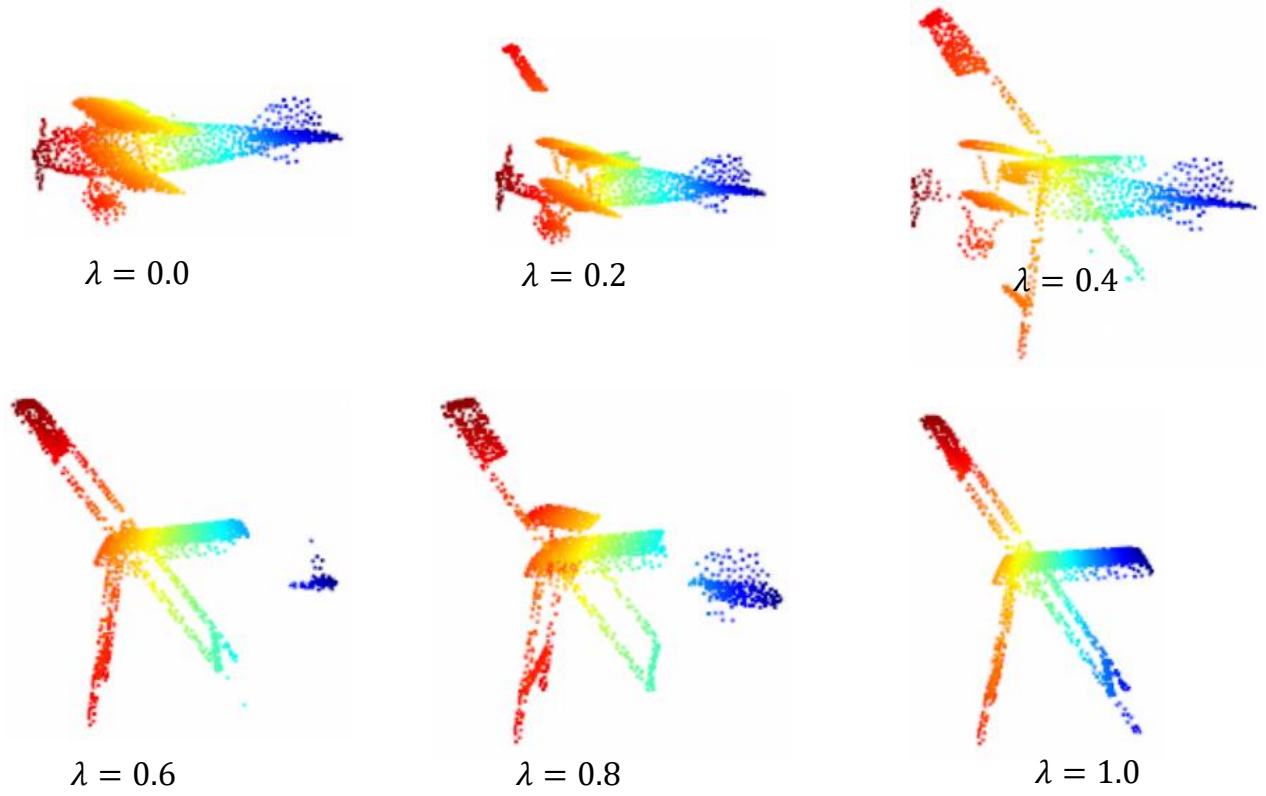


Figure 2.7: The mixed point cloud is visualized by incorporating a plane and a chair using varying replacement ratios λ.

2.5 Summary

Point cloud registration is an essential issue in geomatics, robotics, and computer vision. The goal is to align two or more point clouds representing the same scene from different viewpoints and estimate the transformation that relates them. The applications of point cloud registration are numerous, including 3D reconstruction, object recognition, mapping, navigation, and augmented reality. Despite its importance and potential, point cloud registration is still a challenging task that involves several difficulties. Some of the main challenges are:

- Noise: Point clouds are often affected by noise, which sensor errors, occlusions, or environmental factors can cause. Noise can affect the accuracy of registration and introduce false correspondences.
- Partial overlap: Point clouds may have only a partial overlap, which means that not all points are visible from all viewpoints. This can make it difficult to find matching points and estimate the transformation.
- Large rotation: Point clouds may have a large rotation between them, which can make it difficult to find initial correspondences and converge to a solution.
- Unlabeled data: Point clouds may not have any labels related to correspondences or transformation, which makes it difficult to extract distinguished features to build correspondences.
- Density variations: Point clouds may have variations in density, which can affect the sampling and matching of points.
- Scalability: Point clouds can be very large and complex, which can make it difficult to process and store them efficiently.
- Computational efficiency: Point cloud registration can be computationally expensive, especially for large point clouds or real-time applications.

In conclusion, point cloud registration is a challenging and important task that has many applications in various fields. To enhance the precision and efficacy of point cloud registration, there is a pressing need for extensive research to tackle the issues arising from factors such as noise, partial overlap, extensive rotation, unmarked data, fluctuations in density, scalability, and computational efficiency.

Chapter 3

Data Augmentation-free Unsupervised Learning for 3D Point Cloud Understanding

3.1 Introduction

Section 2.2.1 has discussed that deep learning has successfully solved various 3D point cloud problems. One goal of deep learning is to learn discriminative and transferable point cloud features, which is a crucial problem in the area of 3D shape understanding [172], [173], as it allows efficient training of downstream tasks, ranging from object detection [174] and tracking [175], segmentation [176], reconstruction [177], classification [79] to registration [11], [23], [36]. On the contrary, for neural networks to undergo efficient training, a significant amount of effort is required to manually label data sets. This process involves providing supervisory signals such as point-wise annotations for 3D semantic segmentation and 3D correspondences or poses for 3D point cloud registration. However, annotating point clouds is challenging for several reasons: **(1)** Sparse, low resolution, and irregular spatial distribution of point cloud poses great challenges to annotation [104]; **(2)** The large numbers of points that are contained in point clouds significantly increase the labeling costs and reduce efficiency [104]. Therefore, learning from unlabeled or partially labeled data to alleviate human labeling efforts is an emerging research topic in point cloud understanding as discussed in Section 2.2.1. Along this line, unsupervised representation learning is an attractive yet potent alternative approach to learning features without human intervention [109].

Unsupervised learning approaches can be broadly categorized as generative or discriminative [178]. The former includes self-reconstruction, or auto-encoding [110], generative adversarial

network [179], and auto-regressive [180] methods. These methods can map an input point cloud into a global latent representation [81], [181], or a latent distribution in the variational case [87], [182] through an encoder and then attempt to reconstruct the input by a decoder. Generative methods can effectively model high-level and structural properties of the input point clouds. However, because they are sensitive to Euclidean transformations, they typically assume that all 3D objects have the same pose in a given category [115].

Unlike generative methods, discriminative methods learn to predict or discriminate augmented input versions. These methods can yield rich latent representations for downstream tasks [104]. Examples of these include contrastive methods [81], [115], [183], which have shown remarkable results for unsupervised representation learning. Contrastive methods also promote learning of rotation-invariant representations via data augmentation [184]. It involves training a model to recognize the differences between two or more objects or data points. Typically, these algorithms require several negative samples and heavily depend on the selection criteria to mine negatives [117], [178]. Often, they require large batch sizes, memory banks, or customized strategies to retrieve informative pairs [178]. Moreover, it is unclear what constitutes an effective semantics-preserving data augmentation strategy, which are commonly characterized as a collection of 3D coordinates. It is possible that modifications to the original geometry, such as cropping or occlusion based on viewpoint, could have a negative impact on the overall semantics of the data [109]; for example, random crops of a point cloud may correspond to different objects and introduce inconsistent learning signals. For this reason, contrastive approaches need humans to carefully design combinations of data augmentations for learning informative representations. On the other hand, training on whole object instances can lead to learning global representations, which in turn can produce fewer discriminant representations as local geometric differences may be disregarded [27], [81], [87]. Therefore, the first motivation of this chapter is to design a data augmentation-free, unsupervised learning approach to avoid the inconvenience of building chains of ad-hoc combination data augmentations. The second motivation is to develop an unsupervised method that is not based on global features but can optimize local features, which facilitates the network to learn 3D spatial geometric information of point clouds.

This chapter thus proposes an unsupervised method to learn informative point-level representations of 3D point clouds without data augmentation, named SoftClu. The approach is based on the insightful observation that clustering in both feature and geometric spaces exhibits consistent patterns. Consequently, points within the same cluster in the feature space also tend to have a small distance from each other in the geometric space. This observation enables the formulation of a loss function to train the networks effectively. As shown in Fig. 3.1, the framework learns cluster affiliation scores to softly group the 3D points of each point cloud into a given

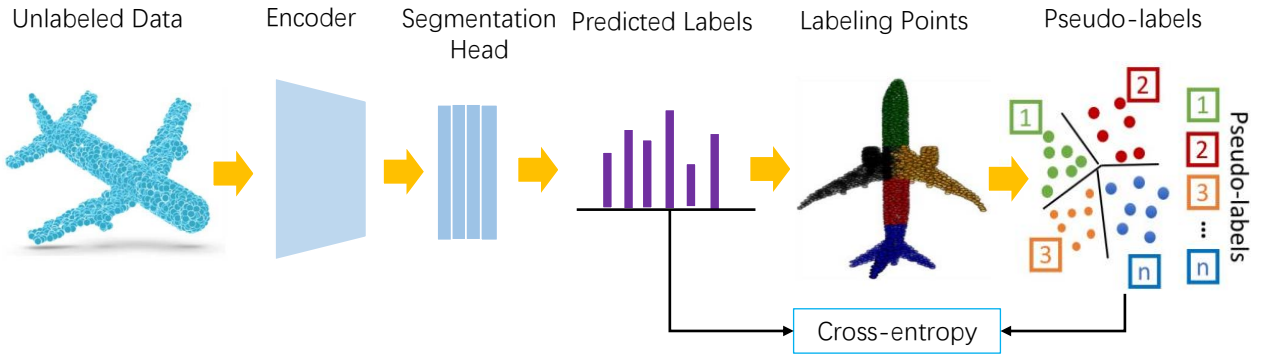


Figure 3.1: The flow diagram of the proposed SoftClu.

number of geometric partitions in coordinate and feature spaces, i.e., through soft clustering. The point-level feature representations are learned by minimizing the standard cross-entropy of a single equation, which is the result of an EM-like algorithm [5]. The Expectation step employs an optimal transport [136] based clustering algorithm to generate point-level pseudo-labels, i.e. focusing on local geometric information. In particular, the method softly labels points based on their distance from the centroids in both feature and geometric spaces, with the constraint that labels partition data in equally-sized subsets. Optimal transport is a potent means for comparing probability distributions with each other and for producing optimal mappings to minimize distances [136]. The Maximization step adapts the E-step for a point-to-cluster loss to optimize the metric learning network. The proposed approach learns the partitioning network itself. SoftClu softly assigns points into geometrically coherent overlapping clusters, overcoming the weakness of conventional GMM and K-means that involve expensive iterative procedures. In doing so, SoftClu avoids data augmentation that might potentially compromise the geometric consistency of the raw point clouds and, consequently, their semantic information. SoftClu is inspired by DeepCluster [185], SeLa [186] and SwAV [187], but it differs from them, as they implement clustering in the feature space at the instance level. They depend on data augmentation and may degrade the geometric information when used with 3D data. Based on the numerical outcomes, it can be observed that utilizing the proposed approach for pre-training on datasets can enhance the performance of various downstream tasks. Furthermore, the results indicate that this method can surpass the current state-of-the-art techniques without any form of data augmentation, even in diverse domains.

To summarize, the contributions of this chapter are:

- This chapter proposes a data augmentation-free unsupervised method, which does not rely on data augmentations, negative pair sampling, and large batches, to learn transferable point-level features on a 3D point cloud;

- This chapter expands the application of pseudo-label prediction to an optimal transport

problem, which can be effectively addressed using a modified version of the Sinkhorn-Knopp algorithm [167];

- This chapter conducts comprehensive experiments, and the approach achieves the best performance without the utilization of data augmentation.

3.2 Methodology

Contrastive methods for unsupervised point cloud representation learning aim to group positive pairs while promoting separation of negative pairs [108], [188]. Differently, representation learning of the proposed unsupervised method implicitly alternates between clustering the point-level features to get point-wise soft-labels (pseudo-labels) and utilizing these soft-labels to train the representations.

The proposed SoftClu formulates the problem of unsupervised representation learning as a soft-clustering problem. Figure 3.2 illustrates its framework, which consists of three steps: prototype computation, soft-label assignment, and optimization. Given a 3D point cloud as an unordered set $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i\}_{i=1}^{N}$ of N points where each point $\boldsymbol{p}_i \in \mathbb{R}^3$ is represented by a 3D coordinate $\boldsymbol{p}_i = \{x, y, z\}$, the goal is to train, in an unsupervised way, a feature encoder f_φ with parameters φ (*e.g.*, PointNet) that extracts informative point-wise features $\boldsymbol{\mathcal{F}} = \{f_\varphi(\boldsymbol{p}_i)\}_{i=1}^{N}$ from $\boldsymbol{\mathcal{P}}$. To this end, SoftClu applies a segmentation head ϕ_α that takes as input $\boldsymbol{\mathcal{F}}$ and outputs joint log probabilities and a softmax operator that acts on log probabilities to generate a classification score matrix $\boldsymbol{S}$. The prototype computation block estimates the J cluster centroids (prototypes) $\boldsymbol{C}^E$ and $\boldsymbol{C}^F$ to represent each partition. Next, soft-labels $\boldsymbol{\gamma}_{ij} \in \boldsymbol{\gamma}$ of each input point $\boldsymbol{p}_i$ are based on these prototypes and the Sinkhorn-Knopp [167] algorithm is employed to perform the soft-label assignment, i.e., $\boldsymbol{\gamma}$ softly groups $\boldsymbol{\mathcal{P}}$ into partitions. $\boldsymbol{\gamma}_{ij} \in [0, 1]$ is a soft-label score that point $\boldsymbol{p}_i$ belongs to cluster j. The final optimization step is to minimize the average cross-entropy loss $\mathcal{L}_{tot}$ between the soft-label $\boldsymbol{\gamma}$ and the predicted category probability $\boldsymbol{S}$. This section is structured in the following manner. Section 3.2.1 outlines the computation of the prototype. Section 3.2.2 presents the derivation of soft-label assignment. Finally, Section 3.2.3 details the optimization procedure.

3.2.1 Prototype computation

SoftClu begins by computing a prototype for each cluster (partition) as the most representative feature for a set of points. Specifically, the point-wise features $\mathcal{F} = \{\boldsymbol{f}_i\}_{i=1}^{N}$, where $\boldsymbol{f}_i = f_\varphi(\boldsymbol{p}_i)$, are utilized to compute a set of classification scores $\boldsymbol{S}$ as

$$\boldsymbol{\mathcal{S}} = \{\sigma(\phi_\alpha(\boldsymbol{f}_i)\}_{i=1}^{N}, \tag{3.1}$$

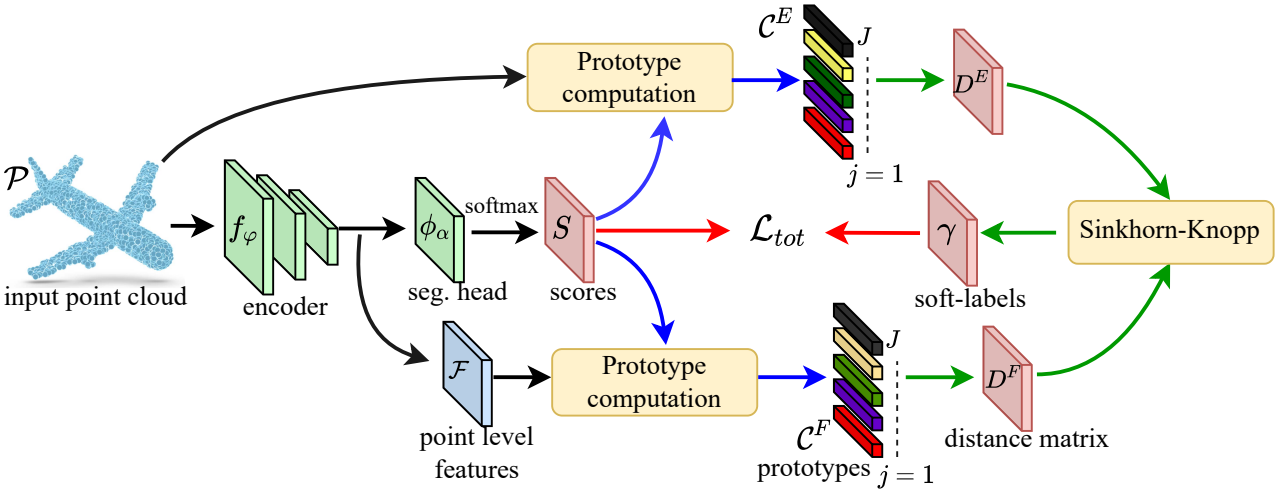


Figure 3.2: The architecture of SoftClu. It consists of three steps: prototype computation (blue line), soft-label $\boldsymbol{\gamma}$ assignment (green line), and optimization (red line).

where σ is the softmax operation, and ϕ_α is the segmentation layer with parameters α. For each feature $\boldsymbol{f}_i$, ϕ_α produces a probability score s_{ij} indicating the likelihood that $\boldsymbol{p}_i$ belongs to partition j. Two types of prototypes are utilized as the representatives for each category, one in the feature space and another in the geometric space. Specifically, according to the type, SoftClu computes J prototypes as the weighted average of the features $\boldsymbol{\mathcal{F}}$ or 3D coordinates $\boldsymbol{\mathcal{P}}$ based on their classification scores $\boldsymbol{S}$. Let $\boldsymbol{C}^E = \{\boldsymbol{c}_j^E\}_{j=1}^J$ be the set of prototypes in the geometric space and let $\boldsymbol{C}^F = \{\boldsymbol{c}_j^F\}_{j=1}^J$ be set of prototypes in the features space that are defined as

$$\boldsymbol{c}_j^E = \frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij} \boldsymbol{p}_i, \tag{3.2}$$

$$\boldsymbol{c}_j^F = \frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij} \boldsymbol{f}_{p_i}. \tag{3.3}$$

3.2.2 Soft-label assignment

The soft-label assignment step labels points based on their distance to the prototypes estimated by Eq. (3.2) and Eq. (3.3). If only features are used for soft-label assignment, this would likely produce disconnected and scattered clusters. Hence, SoftClu concatenates point coordinates with the features so that the label is more localized. Specifically, it encodes the pseudo-labels $\boldsymbol{\gamma} = \{\gamma_{ij} \in [0,1]\}_{i,j}^{N,J}$ as posterior distributions, i.e. soft-labels, satisfying $\sum_{j=1}^J \gamma_{ij} = 1$. γ_{ij} is the posterior probability that $\boldsymbol{p}_i$ belongs to partition j. The proposed method bases the assignment of soft-labels to the respective points on the prototypes $\boldsymbol{C}^E$ and $\boldsymbol{C}^F$, and by following two assumptions:

(i) Cluster cohesion: If a point $\boldsymbol{p}_i$ belongs to partition j, point $\boldsymbol{p}_i$ and prototype $\boldsymbol{c}_j^E$ should

have the shortest distance among the distances of $\boldsymbol{p}_i$ with other prototypes in $\boldsymbol{C}^E$. The same holds true in the feature space.

(ii) Uniform distribution: Each point cloud is assumed to be segmented into equally-sized partitions of $\lfloor \frac{N}{J} \rfloor$ elements, where $\lfloor \cdot \rfloor$ indicates the greatest integer less than or equal to its argument.

Assumption (i) inspires SoftClu to label points based on their distance from the centroids. It can be formalized as an expression, i.e., if $\boldsymbol{p}_i$ belongs to cluster j, then $\|\boldsymbol{p}_i - \boldsymbol{c}_j^E\|_2 \leq \|\boldsymbol{p}_i - \boldsymbol{c}_k^E\|_2, \|\boldsymbol{f}_i - \boldsymbol{c}_j^F\|_2 \leq \|\boldsymbol{f}_i - \boldsymbol{c}_k^F\|_2$ and $\gamma_{ij} \geq \gamma_{ik}, k \neq j, k = 1, \cdots, J$. $\|\cdot\|_2$ is the L2 norm. This can be ensured by minimizing the following objective,

$$\min_{\boldsymbol{\gamma}} \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{J} \left(\lambda \|\boldsymbol{p}_i - \boldsymbol{c}_j^E\|_2^2 + (1-\lambda) \| \boldsymbol{f}_i - \boldsymbol{c}_j^F \|_2^2 \right) \gamma_{ij}, \tag{3.4}$$

where $\lambda \in [0, 1]$ is a learned parameter. For convenience, here defines the following matrix form $\boldsymbol{D} = \lambda \boldsymbol{D}^E + (1-\lambda)\, \boldsymbol{D}^F$, where $\boldsymbol{D}^F = \{\|\boldsymbol{f}_i - \boldsymbol{c}_j^F\|_2^2\}_{i,j}^{N,J}$ and $\boldsymbol{D}^E = \{\|\boldsymbol{p}_i - \boldsymbol{c}_j^E\|_2^2\}_{i,j}^{N,J}$ are matrices of size equal to $N \times J$. Then, Eq. (3.4) can be rewritten as:

$$\min_{\boldsymbol{\gamma}} \left\langle \frac{\boldsymbol{\gamma}}{N}, \boldsymbol{D} \right\rangle, \tag{3.5}$$

where $\langle \cdot, \cdot \rangle$ is the Frobenius matrix dot product.

Assumption (ii) is formulated in a constraint condition as $\sum_{i=1}^{N} \gamma_{ij} = \frac{N}{J}$, which can mitigate the problem that all data points are assigned to a single (arbitrary) label. Therefore, based on $\sum_{i=1}^{N} \gamma_{ij} = \frac{N}{J}$ and the property of the posterior probability $\sum_{j=1}^{J} \gamma_{ij} = 1$, $\boldsymbol{\gamma}$ satisfies the following constraints

$$\begin{aligned} \frac{1}{N} \boldsymbol{\gamma}^\top \mathbf{1}_N &= \frac{1}{J} \mathbf{1}_J, \\ \frac{1}{N} \boldsymbol{\gamma} \mathbf{1}_J &= \frac{1}{N} \mathbf{1}_N, \end{aligned} \tag{3.6}$$

where $\mathbf{1}_k (k = N, J)$ denotes the vector of ones in dimension k.

Let $\boldsymbol{\Gamma} = \frac{\gamma}{N}$ with elements defined as $\Gamma_{ij} = \frac{\gamma_{ij}}{N}$. $\boldsymbol{\Gamma}$ satisfies $\sum_{ij}^{NJ} \boldsymbol{\Gamma}_{ij} = 1$. The optimal transport (OT) problem [136] can be used to formulate the joint objective of assumptions i) and ii) by replacing the variable $\boldsymbol{\gamma}$ with $\boldsymbol{\Gamma}$ in Eq. (3.5) and Eq. (3.6), i.e.,

$$\begin{aligned} &\min_{\boldsymbol{\Gamma}} \langle \boldsymbol{\Gamma}, \boldsymbol{D} \rangle, \\ &\text{s.t. } \boldsymbol{\Gamma}^\top \mathbf{1}_N = \frac{1}{J} \mathbf{1}_J, \boldsymbol{\Gamma} \mathbf{1}_J = \frac{1}{N} \mathbf{1}_N. \end{aligned} \tag{3.7}$$

The minimization of Eq. (3.7) can be solved in polynomial time as a linear program. However, the linear program involves millions of data points and thousands of classes and traditional algorithms hardly scale to large problems [167]. This issue can be addressed by adopting an

efficient version of the Sinkhorn-Knopp algorithm [167]. The implementation of the Sinkhorn-Knopp algorithm is described in Alg. 2. This requires the following regularization term

$$\begin{aligned} &\min_{\boldsymbol{\Gamma}} \langle \boldsymbol{\Gamma}, \boldsymbol{D} \rangle - \epsilon H(\boldsymbol{\Gamma}), \\ &\text{s.t. } \boldsymbol{\Gamma}^\top \mathbf{1}_N = \frac{1}{J}\mathbf{1}_J, \;\; \boldsymbol{\Gamma}\mathbf{1}_J = \frac{1}{N}\mathbf{1}_N, \end{aligned} \tag{3.8}$$

where $H(\boldsymbol{\Gamma}) = \langle \boldsymbol{\Gamma}, \log \boldsymbol{\Gamma} - 1 \rangle$ denotes the entropy of $\boldsymbol{\Gamma}$ and $\epsilon > 0$ is a regularization parameter. For very small ϵ, optimizing Eq. (3.8) is equivalent to optimizing Eq. (3.7), but even for moderate values of ϵ, the objective tends to have approximately the same optimizer [167]. The larger the ϵ, the faster the convergence, please refer to [167] for details. In this chapter, using a fixed $\epsilon = 1e-3$ is appropriate as SoftClu is only interested in the final clustering and representation learning results, rather than in solving the transport problem exactly. The solution to Eq. (3.8) takes the form of the following normalized exponential matrix [167],

$$\boldsymbol{\Gamma} = \operatorname{diag}(\boldsymbol{\mu}) \exp\left(\boldsymbol{D}/\epsilon\right) \operatorname{diag}(\boldsymbol{\nu}), \tag{3.9}$$

where $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots, \mu_N)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \cdots, \nu_J)$ are renormalization vectors in $\mathbb{R}^N$ and $\mathbb{R}^J$. The vectors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ can be obtained by iterating the updates via $\boldsymbol{\mu}_i = \left[\exp\left(\boldsymbol{D}/\epsilon\right)\boldsymbol{\nu}\right]_i^{-1}$ and $\boldsymbol{\nu}_j = \left[\exp\left(\boldsymbol{D}/\epsilon\right)^\top \boldsymbol{\mu}\right]_j^{-1}$ with initial values $\boldsymbol{\mu} = \frac{1}{N}\mathbf{1}_N$ and $\boldsymbol{\nu} = \frac{1}{J}\mathbf{1}_J$, respectively. The initialization of $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ can be any distribution, and choosing the constraints as initial values allow a faster convergence [167]. $[\cdot]_j^{-1}$ defines as the inverse value of the j^{th} element of its argument. In all experiments, 20 iterations are used as it works well in practice. After solving Eq. (3.9), the soft-label matrix can be inferred as

$$\boldsymbol{\gamma} = N \cdot \boldsymbol{\Gamma}. \tag{3.10}$$

3.2.3 Optimization

The optimization step follows an EM-like scheme where the Expectation step E optimizes prototypes and soft labels, while the Maximization step M optimizes the trained parameters for representation learning. Each step can be detailed as follows:

- E: Given the current encoder and segmentation layer, SoftClu computes prototypes $\boldsymbol{C}^{\boldsymbol{E}}$ and $\boldsymbol{C}^F$ following Eq. (3.2) and Eq. (3.3), and obtain soft-labels $\boldsymbol{\gamma}$ through $\boldsymbol{\gamma} = N \cdot \boldsymbol{\Gamma}$.
- M: Given the current soft-labels $\boldsymbol{\gamma}$ from step E, SoftClu optimizes the encoder f_φ and segmentation layer ϕ_α parameters.

During E, SoftClu solves the OT problem with the Sinkhorn-Knopp algorithm. During M, SoftClu minimizes the segmentation loss based on the resulting soft labels, such as

$$\mathcal{L}_{soft}(\boldsymbol{\gamma}, \boldsymbol{S}) = -\frac{1}{N}\langle \boldsymbol{\gamma}, \log \boldsymbol{S} \rangle = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{J} \boldsymbol{\gamma}_{ij} \log s_{ij}, \tag{3.11}$$

which corresponds to the minimization of the standard cross-entropy loss between soft-labels $\boldsymbol{\gamma}$ and predictions $\boldsymbol{S}$. However, the optimization of Eq. (3.11) does not ensure encoder f_φ from predicting the same features for all the points, i.e. all centroids collapse into the same vector. SoftClu further promotes centroids separation by minimizing the orthogonal regularization loss

$$\mathcal{L}_{orth}(\boldsymbol{C}) = \|\boldsymbol{C}^{E}{}_{*}^{\top}\boldsymbol{C}_{*}^{E} - \boldsymbol{I}\|_{Fr} + \|\boldsymbol{C}^{F}{}_{*}^{\top}\boldsymbol{C}_{*}^{F} - \boldsymbol{I}\|_{Fr}, \tag{3.12}$$

where $\boldsymbol{C}_{*}^{k} = [\frac{\boldsymbol{c}_1^k}{\|\boldsymbol{c}_1^k\|_2}, \frac{\boldsymbol{c}_2^k}{\|\boldsymbol{c}_2^k\|_2}, \cdots, \frac{\boldsymbol{c}_J^k}{\|\boldsymbol{c}_J^k\|_2}]$ with $k = E, F$, and $\|\cdot\|_{Fr}$ is Frobenius norm. The final objective loss for the M step is defined as

$$\mathcal{L}_{tot} = \mathcal{L}_{soft} + \eta\mathcal{L}_{orth}, \tag{3.13}$$

where $\eta = 0.01$ is a weighting parameter. This chapter sets the value of η empirically and finds that $\eta \leq 0.01$ can slightly improve the performance. The minimization of this loss leads to the maximization of the expected number of points correctly classified, associating the correct neighbor prototypes. This facilitates the encoder to learn more local geometric information. The implementation of SoftClu is described in Alg. 1.

Algorithm. This section provides a pseudo-code for SoftClu training loop in Algorithm 1. For the Sinkhorn-Knopp algorithm, a detailed pseudo-code is provided in Algorithm 2.

3.3 Experiments

The following section outlines the pre-training and downstream fine-tuning procedures discussed in Sec.3.3.1 and Sec.3.3.2, respectively. To evaluate the effectiveness of the proposed method, three scenario setups, namely object classification, 3D part, and semantic segmentation, are utilized and discussed in Sec.3.3.3. Finally, Sec.3.3.4 includes an ablation study and experimental analysis.

3.3.1 Pre-training setup

SoftClu was implemented in PyTorch and executed experiments on two Tesla V100-PCI-E-32G GPUs. The cluster settings for pre-training were set to $J = 64$ and $\epsilon = 1e-3$ as they have been found to work effectively in practice. The segmentation head ϕ_α is composed of three fully connected layers, each consisting of a linear layer followed by batch normalization. The hidden layer and final linear layer output dimensions are half the dimensions of the encoder output and the number of clusters, respectively. All layers, except for the final layer, have a rectified linear unit. To evaluate the efficacy of the proposed method on various downstream tasks, Point-based PointNet [79], graph-based DGCNN [189], full convolution-based SR-UNet provided in PointContrast [27], and Transformer-based encoder [97] provided by MaskPoint [190]

Algorithm 1 Soft clustering (pseudocode).

Input: $\{\mathcal{P}\}$ a set of 3D point clouds and each point cloud has N points; K number of optimization steps.

Output: the backbone f_φ pretrained by using the proposed algorithm.

for i in range(0, K) **do**
  $\mathcal{L}_{tot} = 0$
  for $\mathcal{P} \in \{\mathcal{P}\}$ **do**
    # compute class scores
    $\boldsymbol{S} = \text{softmax}\left(\phi_\alpha\left(f_\varphi\left(\mathcal{P}\right)\right)\right)$
    # compute prototypes
    $\boldsymbol{C}^E = \left\{\frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij}\boldsymbol{p}_i\right\}_{j=1}^N$
    $\boldsymbol{C}^F = \left\{\frac{1}{\sum_{i=1}^N s_{ij}} \sum_{i=1}^N s_{ij}\boldsymbol{f}_i\right\}_{j=1}^N$
    # compute $\boldsymbol{D}$
    $\boldsymbol{D} = \left\{\lambda\|\boldsymbol{p}_i - \boldsymbol{c}_j^E\|_2^2 + (1-\lambda)\|\boldsymbol{f}_i - \boldsymbol{c}_j^F\|_2^2\right\}_{i,j}^{N,J}$
    # compute γ
    $\boldsymbol{\Gamma} = \text{SINKHORN}\left(\text{stopgrad}\left(\boldsymbol{D}\right), 1e-3, 20\right)$
    $\boldsymbol{\gamma} = N \cdot \boldsymbol{\Gamma}$
    # compute loss
    $\mathcal{L}_{tot}$ += $\mathcal{L}_{soft} + \eta\mathcal{L}_{orth}$
  end for
  # update backbone and segmentation head
  $f_\varphi, \phi_\alpha \leftarrow \text{optimize}\left(\frac{\mathcal{L}_{tot}}{N}\right)$
end for
return f_φ

was chosen as backbones. PointNet extracts features after the point-wise max-pool operation, and DGCNN collects features after pooling the fifth EdgeConv layer. For Transformer, feature PointNet++ [82] propagation is first performed as SoftClu relies on pointwise features, independently upsampling the downsampled point clouds into the number of points of the input point cloud. This section assesses pre-training strategies on complex scenes with single objects (ShapeNet [191]) and multiple objects (ScanNet [192]) to examine the effectiveness of SoftClu.

ShapeNet [191] is a dataset consisting of CAD models of individual objects, with over 50,000 synthetic objects from 55 categories. Following [108], each point cloud is randomly sampled to 2048 points. The official training split of ShapeNet is used for pre-training. Object classification, part segmentation, semantic segmentation, and few-shot classification use the pre-training models on ShapeNet to evaluate the performance of SoftClu. SoftClu pre-trained

Algorithm 2 Sinkhorn-Knopp algorithm (pseudocode).

Input: D distance matrix, $\epsilon = 1e-3$ and $niters$ iterations.

function SINKHORN($\boldsymbol{D}$, ϵ, $niters$)
    $\boldsymbol{\Gamma} = \exp(\boldsymbol{D}/\epsilon)$
    $\boldsymbol{\Gamma} / = \text{sum}(\boldsymbol{\Gamma})$
    $N, J = \boldsymbol{\Gamma}.\text{shape}$
    $\boldsymbol{u}, \boldsymbol{\mu}, \boldsymbol{\nu} = \text{zeros}(N), \text{ones}(N)/N, \text{ones}(J)/J$
    for _ in range(0, $niters$) **do**
        $\boldsymbol{u} = \text{sum}(\boldsymbol{\Gamma}, \text{dim=1})$
        $\boldsymbol{\Gamma} * = (\boldsymbol{\mu}/\boldsymbol{u}).\text{unsqueeze}(1)$
        $\boldsymbol{\Gamma} * = (\boldsymbol{\nu}/\text{sum}(\boldsymbol{\Gamma}, \text{dim=0})).\text{unsqueeze}(0)$
    end for
return $\boldsymbol{\Gamma}$
end function

PointNet and DGCNN on the ShapeNet dataset, with both encoders using a latent dimension of 1024. Pre-training involved 250 epochs using the AdamW [193] optimizer with a batch size of 32 and an initial learning rate of 0.001, which decayed by 0.7 every 20 epochs.

ScanNet [192] is a dataset of indoor scenes with multiple objects and consists of 1513 reconstructed meshes for 707 unique scenes. Each point cloud is sampled to 4096 points using a fast point sampling algorithm. SR-UNet was pre-trained on ScanNet using an SGD optimizer with a learning rate of 0.1 and a batch size of 32. The learning rate decreases by a factor of 0.99 for every 1K iteration. The model is trained for 30K iterations.

3.3.2 Downstream tasks setups

SoftClu is evaluated using four downstream tasks, namely classification, part segmentation, semantic segmentation, and few-shot learning, to assess its performance. The comparison is made against other discriminative approaches that are currently considered state-of-the-art (Jigsaw3D [116], STRL [108], CrossPoint [194], SimCLR [188], STRL [108], PointContrast [27], and ContrastiveScene [195]) and two generative approaches (OcCo [104] and ParAE [109]).

Linear SVM classification. Linear SVM classification on ModelNet40 [196] and ModelNet10 [196] datasets are used to evaluate the quality of their pre-trained versions on ShapeNet. ModelNet40 consists of 12331 meshed models from 40 object categories, divided into 9843 training meshes and 2468 testing meshes, with points sampled from CAD models. ModelNet10 dataset comprises 4899 pre-aligned shapes from 10 categories, with 3991 (80%) shapes for train-

ing and 908 (20%) shapes for testing. For SVM training, each point cloud is randomly sampled to 1024 points from each shape, following [116]. This classifying task uses the frozen encoders to extract point-level features, followed by max-pooling to generate global feature vectors, and trains a linear SVM for classification.

Part segmentation. Part segmentation involves predicting labels for each point based on its corresponding part and requires a thorough understanding of local patterns. Following [104], ShapeNetPart [197] is chosen as a benchmark to evaluate the part segmentation performance. ShapeNetPart contains 16881 objects of 2048 points from 16 categories with 50 parts in total. This setup fine-tunes the encoders with a linear fully connected layer for 100 epochs using the AdamW [193] optimizer, with a batch size of 24, an initial learning rate of 0.001, and a decay rate of 0.5 every 20 epochs. The overall accuracy (OA) and the mean class intersection over union (mIoU) are adopted to evaluate segmentation quality as in [104].

Semantic segmentation. Semantic segmentation on large-scale 3D scenes presents a challenge, as it requires an understanding of contextual semantics and local geometric relationships. To assess the suitability of pre-trained features for the task at hand, the S3DIS benchmark dataset [198] was utilized. This dataset comprises 3D scans of six indoor areas with a total of 271 rooms and 13 semantic classes. The data was collected using the Matterport scanner. Prior to analysis, each room was divided into $1m \times 1m$ blocks, and 4,096 points were used as model input, in accordance with the approach taken in [104]. Fine-tuning of each encoder was conducted on areas 1-4 and 6, with testing, carried out on area 5. The SGD optimizer was employed, with a momentum of 0.9 and a weight decay of 1e-4 for PointNet and DGCNN. The learning rate decayed using cosine annealing with a minimum value of 1e-3, and the models were trained over 250 epochs with a batch size of 48. The segmentation quality was assessed using OA and mIoU metrics. The SR-UNet backbone was also used, with training conducted using a batch size of 48 over 10K iterations. The initial learning rate was set at 0.1, and polynomial decay with a power of 0.9 was used. A voxel size of 0.05 (5cm) was employed, with a weight decay of 0.0001. Segmentation quality was evaluated using mAcc and mIoU metrics, as conducted in [27].

Few-shot learning. This experiment focuses on Few-shot learning (FSL) and its ability to train a model with limited data for the classification task. Specifically, the experiment applies FSL (N-way K-shot learning) on the benchmark datasets ModelNet40 [196] and ModelNet10 [196], where the model is evaluated on N classes, with each category containing N samples. The train/test splits of datasets are consistent with OcCo [104] and CrossPoint [194], and the results are reported as the mean and standard deviation across 10 runs.

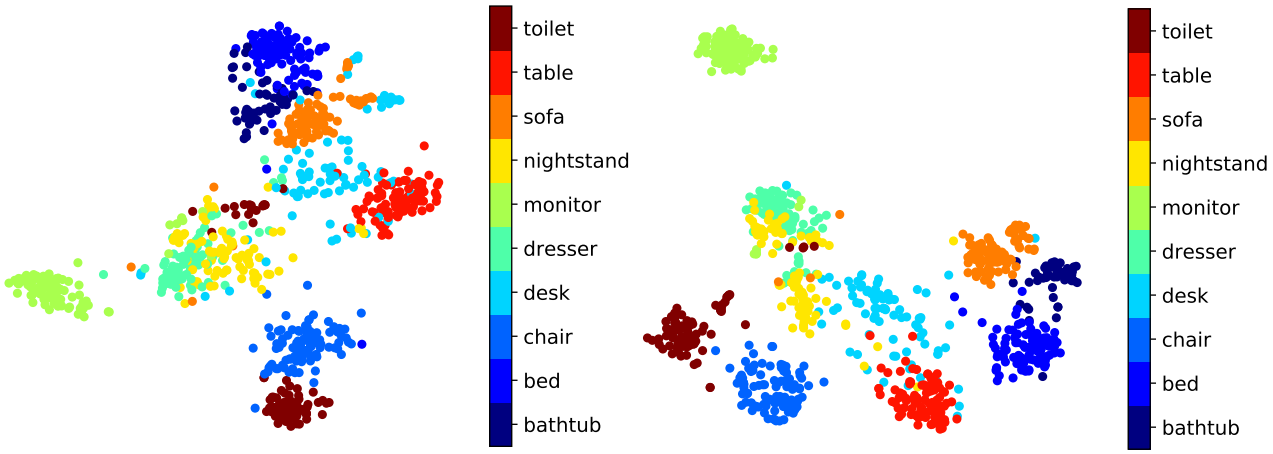
Table 3.1: Linear SVM classification comparisons on ModelNet40 and ModelNet10. ⋆ indicates that models are pre-trained on ModelNet40, otherwise, models are pre-trained on ShapeNet. using Bold font indicates the best performance

Method	Year	PointNet		DGCNN	
		ModelNet40	ModelNet10	ModelNet40	ModelNet10
DeepCluster [185]	2018	86.3	91.6	90.4	94.1
Jigsaw3D⋆ [116]	2019	87.5	91.3	87.8	92.6
Jigsaw3D [116]	2019	87.3	91.6	90.6	94.5
Rotation3D [184]	2020	88.6	-	90.8	-
SwAV [187]	2020	85.4	92.1	90.3	93.5
OcCo [104]	2021	88.7	91.4	89.2	92.7
SimCLR [188]	2021	88.4	91.4	90.1	92.1
STRL [108]	2021	88.3	-	90.9	-
SimSiam [199]	2021	88.7	92.4	91.2	93.8
ParAE [109]	2021	**90.3**	-	91.6	-
CrossPoint [194]	2022	89.1	-	91.2	-
SoftClu⋆	-	88.4	93.0	91.4	94.5
SoftClu	-	**90.3**	**93.5**	**91.9**	**94.8**

3.3.3 Downstream fine-tuning results

Linear SVM classification. Table 3.1 reports the classification accuracy of the proposed SoftClu, compared to the current outstanding baselines. Results show that SoftClu is more effective than the alternative pre-training methods on both datasets. Specifically, on ModelNet40, the proposed SoftClu with PointNet backbone achieves the same classification accuracy (90.3%) as ParAE [109] while outperforming the contrastive approach STRL [108] (88.3%). The linear SVM classification performance of SoftClu even surpasses the fully supervised PointNet, achieving an 89.2% test accuracy. With the DGCNN encoder, SoftClu achieves a 91.9% test accuracy, outperforming ParAE (91.6%) by 0.3%, and generating model OcCo [104] by 2.7%. On ModelNet10, SoftClu outperforms OcCo [104] and Jigsaw3D [116] with both encoding networks. Compared to jigsaw tasks that coarsely segment a point cloud into disjoint partitions, SoftClu learns the partitioning function to assign point clouds into coherent clusters softly. Moreover, this experiment also reports results (SoftClu⋆) of SoftClu pre-trained on ModelNet40 since some methods are pre-trained on that dataset. One can note that SoftClu also performs well and achieves competitive results. Specifically, when used with PointNet, SoftClu achieves an accuracy of 88.4% and 93.0% on ModelNet40 and ModelNet10, respectively. On the other hand, when using the DGCNN backbone, SoftClu achieves an accuracy of 91.4% and

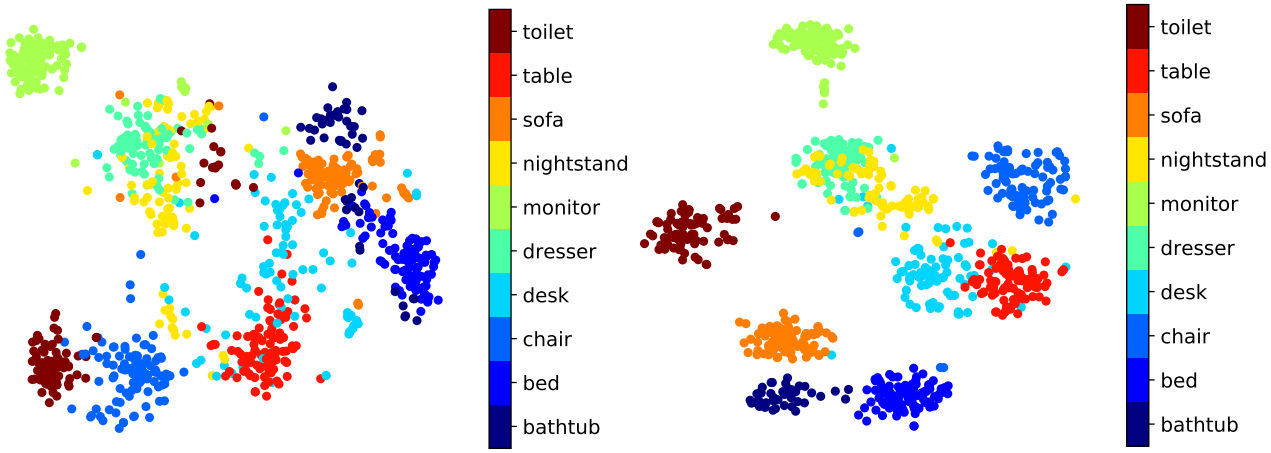
94.5% on ModelNet40 and ModelNet10, respectively. This experiment further uses visualization to explore pre-trained features before fine-tuning with the DGCNN encoder. The feature representation of OcCo and SoftClu on ModelNet10 using PointNet and DGCNN backbones are visualized with T-SNE in Figs. 3.3 and 3.4, respectively. SoftClu displays a superior feature separation compared to OcCo, indicating its greater ability to cluster objects in the feature space. In Fig. 3.5, the points are color-coded based on the PCA projections of the network features, highlighting the pre-trained encoder's effectiveness in embedding geometric information.



(a) OcCo [104] (b) SoftClu

Figure 3.3: T-SNE visualizations of the unsupervisedly learned representations using PointNet backbone on the ModelNet10 test set. Different color represents different categories. (a) OcCo [104] and (b) SoftClu. The proposed SoftClu produces better separated and grouped clusters for different categories.

Part segmentation. Table 3.2 reports the part segmentation results of SoftClu in comparison with the alternative approaches on ShapeNetPart [197]. SoftClu outperforms all the other approaches with both PointNet and DGCNN encoders in terms of both OA and mIoU. With the PointNet encoder, SoftClu achieves 93.9% OA and 83.8% mIoU, improving over state-of-the-art CrossPoint (93.2% OA, 82.7% mIoU) by 0.7% OA and 1.1% mIoU. With the DGCNN encoder, it also achieves 94.6% OA and 85.7% mIoU, outperforming CrossPoint (94.4 OA, 85.3% mIoU) of about 0.2% OA and 0.4% mIoU. Figure 3.6 shows examples of qualitative part segmentation results obtained with SoftClu after the fine-tuning on the downstream task compared to ground-truth annotations (GT). It can be observed that SoftClu provides consistent predictions throughout shapes, also in the case of complex shapes (chairs and motorcycles).



(a) OcCo [104] (b) SoftClu

Figure 3.4: T-SNE visualizations of the unsupervisedly learned representations using DGCNN backbone on the ModelNet10 test set. Different color represents different categories. (a) OcCo [104] and (b) SoftClu. The proposed SoftClu produces better separated and grouped clusters for different categories.

Table 3.2: Part segmentation results on the ShapeNetPart dataset using the pre-trained PointNet and DGCNN backbones. The bold font indicates the best performance.

Method	PointNet		DGCNN	
	OA (%)	mIoU (%)	OA (%)	mIoU (%)
Random	92.8	82.2	92.2	84.4
Jigsaw3D [116]	93.1	82.2	92.7	84.3
OcCo [104]	93.4	83.4	94.4	85.0
CrossPoint [194]	93.2	82.7	93.2	82.7
SoftClu (Ours)	**93.9**	**83.8**	**94.6**	**85.7**

Semantic segmentation. Table 3.3 reports the segmentation results of SoftClu and that of the other baselines on S3DIS [198]. SoftClu outperforms all the other approaches with both PointNet and DGCNN encoders. With the PointNet encoder, SoftClu achieves 82.9% OA and 55.3% mIoU, outperforming both the state-of-the-art OcCo (82.0% OA, 55.3% mIoU) and Jigsaw3D (80.1% OA, 52.6% mIoU). With the DGCNN encoder, SoftClu achieves 85.4% OA and 59.2% mIoU, outperforming CrossPoint [194] (84.7% OA and 58.4% mIoU), OcCo (84.6% OA, 58.0% mIoU) and Jigsaw3D (84.1% OA, 55.6% mIoU). This experiment also compares SoftClu with point-level methods such as PointContrast [27] and ContrastiveScene [195] when features are pre-trained on ScanNet with SR-UNet backbone. Table 3.4 shows that the proposed SoftClu achieves 73.4% mIoU and 79.1% mAcc, outperforming the pre-training results

Figure 3.5: SoftClu allows for unsupervised learning of point-level representations without using data augmentation. These representations embed rich geometric information for point cloud classification and segmentation tasks.

Table 3.3: 3D semantic segmentation mIoU results on the S3DIS dataset using different pre-trained backbones.

Method	PointNet		DGCNN	
	OA (%)	mIoU (%)	OA (%)	mIoU (%)
Random	78.9	47.0	83.7	54.9
Jigsaw3D [116]	80.1	52.6	84.1	55.6
OcCo [104]	82.0	54.9	84.6	58.0
CrossPoint [194]	81.8	54.5	84.7	58.4
SoftClu (Ours)	**82.9**	**55.3**	**85.4**	**59.2**

of PointContrast [27] and ContrastiveScene [195]. It further shows the effectiveness of the proposed method, even on point clouds with multiple objects.

Table 3.4: Results of semantic segmentation with SR-UNet backbone [198].

Method	Scratch	PointContrast [27]	ContrastiveScene [195]	SoftClu
mIoU	68.2	70.3	72.2	**73.4**
mAcc	75.5	76.9	-	**79.1**

Few-shot learning. The FSL results on ModelNet40 are presented in Table 3.5. It is observed that SoftClu outperforms prior works in all the FSL settings with the DGCNN backbone,

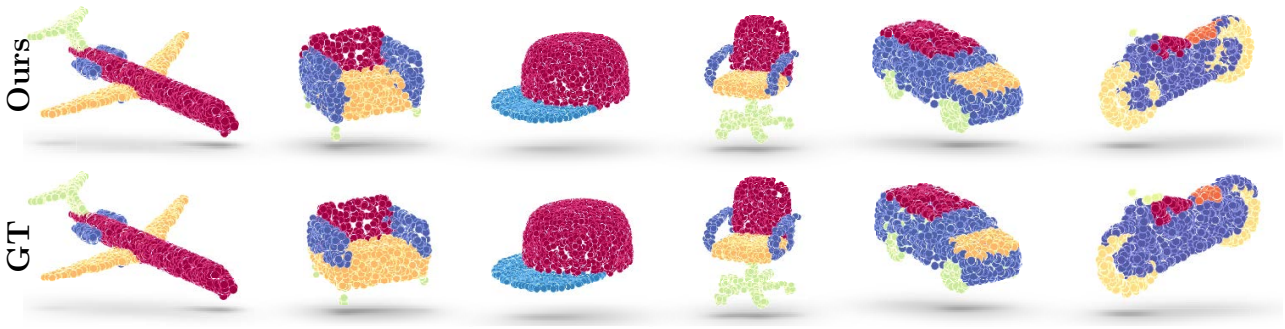


Figure 3.6: Part segmentation results on ShapeNetPart [197] of SoftClu using the DGCNN encoder (top row) compared to the ground-truth labels (bottom row).

and it significantly outperforms the second-best method, CrossPoint. However, the proposed SoftClu with PointNet backbone exhibits slightly poorer performance in the 5-way 10-short and 10-way 20-short settings compared to CrossPoint with PointNet. These results demonstrate that the proposed approach can offer a valuable initialization for downstream tasks, as it can incorporate rich geometric information into the extracted point cloud features.

Transformer backbone. Following [190] setups, this section also provides the results with the recent Transformer backbone provided by [190] to further explore the effectiveness of SoftClu. Finally, clustering is applied to train the Transformer. The Transformer encoder will be used with their pre-trained weights as initialization for the classification task. Tab. 3.6 reports the classification results with a Transformer backbone on ModelNet40, and the proposed method achieves a competitive result.

3.3.4 Ablation study and analysis

Cluster numbers. In this experiment, the effect of selecting different numbers of cluster partitions J is first explored using ModelNet40. SoftClu is pre-trained with different values of J, ranging from 16 to 128, and the results are reported in Table 3.7. The best results are achieved with $J = 64$ for both PointNet and DGCNN. It can be observed that the results vary slightly across different values of J. This suggests that the number of clusters has little influence as long as there are sufficient clusters.

Feature and geometric prototypes. This experiment aims to investigate the impact of feature and geometric prototypes on the performance of SoftClu (Eq. 3.2). Three different pre-training methods are employed: (i) using only feature prototypes, (ii) using only geometric prototypes, and (iii) using both prototypes on ModelNet40 and ModelNet10 datasets. According to the results in Table 3.8, SoftClu with only feature prototypes achieves the lowest performance. This can be attributed to the misclassification of points that have similar features

Table 3.5: The results of few-shot object classification on ModelNet40 are presented, showing the mean and standard error over 10 runs. The top-performing results for each backbone are indicated in bold.

Encoder	Method	5-way		10-way	
		10-shot	20-shot	10-shot	20-shot
PointNet	Rand	52.0 ± 3.8	57.8 ± 4.9	46.6 ± 4.3	35.2 ± 4.8
	Jigsaw [116]	66.5 ± 2.5	69.2 ± 2.4	56.9 ± 2.5	66.5 ± 1.4
	cTree [200]	63.2 ± 3.4	68.9 ± 3.0	49.2 ± 1.9	50.1 ± 1.6
	OcCo [104]	89.7 ± 1.9	92.4 ± 1.6	83.9 ± 1.8	89.7 ± 1.5
	CrossPoint [194]	**90.9 ± 4.8**	93.5 ± 4.4	84.6 ± 4.7	**90.2 ± 2.2**
	SoftClu	90.6 ± 4.0	**93.8 ± 3.2**	**84.7 ± 3.6**	90.1 ± 4.5
DGCNN	Rand	31.6 ± 2.8	40.8 ± 4.6	19.9 ± 2.1	16.9 ± 1.5
	Jigsaw [116]	34.3 ± 1.3	42.2 ± 3.5	26.0 ± 2.4	29.9 ± 2.6
	cTree [200]	68.4 ± 3.4	71.6 ± 2.9	42.4 ± 2.7	43.0 ± 3.0
	OcCo [104]	90.6 ± 2.8	92.5 ± 1.9	82.9 ± 1.3	86.5 ± 2.2
	CrossPoint [194]	92.5 ± 3.0	94.9 ± 2.1	83.6 ± 5.3	87.9 ± 4.2
	SoftClu	**93.6 ± 3.3**	**97.3 ± 2.0**	**89.1 ± 1.4**	**93.2 ± 3.4**

Table 3.6: Classification results with a Transformer backbone on ModelNet40.

Encoder	SoftClu	PointViT-OcCo [104]	Point-BERT [118]	MaskPoint [190]
SoftClu	**93.8**	92.1	93.2	**93.8**

Table 3.7: Ablation study results of SoftClu with different number of clusters J.

Method	16	32	48	64	72	96	112	128
PointNet	92.4	93.0	93.1	93.5	93.4	93.3	93.2	93.1
DGCNN	94.2	94.8	94.6	94.8	94.7	94.6	94.6	94.5

but belong to different geometric regions, such as the wings of an airplane. In contrast, SoftClu with both feature and geometric prototypes achieves the best performance on both ModelNet40 and ModelNet10, when combined with both PointNet and DGCNN.

Batch size. Contrastive methods that utilize negative examples from mini-batches can face a decline in performance when the batch size is small, as reported in [178]. In contrast, SoftClu is more robust to smaller batch sizes as it does not rely on negative examples. To demonstrate this, an experiment was conducted to compare the performance of SoftClu with SimCLR [188]

Table 3.8: Ablation study of SoftClu by using different prototypes.

Encoder	Geometry	Feature	Accuracy	
			ModelNet40	ModelNet10
PointNet	✓		88.7	92.9
		✓	86.5	92.7
	✓	✓	**90.3**	**93.5**
DGCNN	✓		91.4	94.5
		✓	90.5	93.3
	✓	✓	**91.9**	**94.8**

Table 3.9: Ablation study results of SoftClu by using DGCNN on ModelNet10 with different batch sizes during pre-training.

Encoder	Method	8	16	24	32	40	48
PointNet	SimCLR	87.5	88.0	88.2	88.1	88.5	88.4
	SoftClu	89.9	89.8	90.2	90.3	90.1	89.9
DGCNN	SimCLR	88.6	89.3	89.4	89.7	89.7	90.1
	SoftClu	91.6	91.8	91.7	91.9	91.9	91.8

under different batch sizes ranging from 8 to 48 during pre-training. The results are shown in Table 3.9, which clearly demonstrate that SimCLR experiences performance degradation when the batch size is set to 8, most likely due to the inadequate number of negative samples. On the other hand, SoftClu maintains a stable performance across different batch size configurations.

Computation of soft-labels. In this experiment, the performance of the optimal transport (OT) based soft-label assignment strategy is compared with a typical L2 distance-based approach on ModelNet40 and ModelNet10. SoftClu is evaluated using Γ computed with Eq.(3.9) and the L2 approach in [185]. Results in Table 3.10 show that OT outperforms the L2 approach on all datasets with both PointNet and DGCNN encoders. This is attributed to the equal partition constraint in Alg. 2, which ensures that solutions are not assigned to the same cluster, thereby improving performance.

Running times. The SoftClu pre-training approach is utilized exclusively, with each iteration involving two components: a backbone forward pass and SoftClu optimization. The iteration time was gauged over numerous iterations, with SoftClu executed on one Tesla V100 GPU (32G) and two Intel(R) 6226 CPUs. For ShapeNet, SoftClu results in an average overhead of

Table 3.10: Ablation study of SoftClu on ModelNet40 and ModelNet10 with soft-labels computed with the proposed approach (OT) and with a typical distance-based assignment (L2).

Dataset	Encoder	Accuracy	
		L2	OT
ModelNet10	PointNet	91.5	**93.4**
	DGCNN	94.1	**94.8**
ModelNet40	PointNet	86.5	**90.3**
	DGCNN	90.4	**91.9**

0.014ms for each iteration using a DGCNN backbone. It is noteworthy that the inference time for each backbone remains unchanged, as SoftClu is not involved again.

3.4 Summary and Conclusions

This chapter has introduced SoftClu, a novel unsupervised representation learning approach for understanding 3D point clouds, which does not require data augmentation. SoftClu works by iteratively clustering point-level features to generate pseudo-labels, which are then used to train the representations. The results of this method have been promising, demonstrating its ability to transfer pre-trained representations to various 3D understanding tasks, such as semantic segmentation, classification, and part segmentation. Moreover, SoftClu is highly flexible, as it is not dependent on any specific deep network architecture, and can be used as a pre-training method to improve the performance of other 3D models by extracting distinguishable features from raw point cloud data. While the current chapter does not present any results related to point cloud registration, SoftClu can also be applied to registration tasks. The following chapter will delve into the use of the SoftClu to enhance registration tasks.

Chapter 4

Unsupervised Point Cloud Registration with Beam Search and Soft Segmentation

4.1 Introduction

Chapter 3 provided a way to learn point cloud features without label information, unsupervised learning. This chapter will further discuss applying the unsupervised method to point cloud registration. There has been a growing interest in developing robust and efficient registration algorithms. Mainly, deep learning techniques have been applied to facilitate rigid alignment algorithms greatly. Compared to non-learning-based registration, deep learning-based methods have the advantages of being fast, high accuracy, and robust to noise. Learning-based approaches can be classified into two categories, correspondence-based and correspondences-free [3]. The main idea of the former is to use the extracted per-point or per-patch deep features to estimate correct correspondences. Then, optimization algorithms utilize the correspondences to calculate the rigid transformation. Examples range from DCP [10], RPMNet [133] and Deep-GMR [16] to DGR [12]. However, these approaches require correspondence searching, which is time-consuming and sensitive to large rotation, density variation, and outliers. Besides, most correspondence-based approaches rely on point-point correspondence information to obtain reliable point-wise features for point cloud registration in the training stage. The label information is either inaccessible sometimes or expensive in the human annotation. Correspondence-free methods, such as PointNetLK [1], regress the rigid motion parameters by minimizing the difference between the global features of two input point clouds. The key point is that the extracted global features must be sensitive to the pose [3]. Such methods do not require point-point correspondences and are robust to density variation. Besides, these methods use extracted global

features to reduce the point cloud's dimension so that the algorithm's time complexity does not increase as the number of points grows.

However, despite their utility, these methods still exhibit several limitations. Firstly, their performance in handling large rotations needs improvement. Secondly, akin to deep learning, they rely on labeled data for training feature extractors. Lastly, their effectiveness is contingent upon significant overlaps between the two point clouds, leading to a decline in performance when dealing with partially overlapped point clouds.

To overcome these limitations, this chapter introduces a novel point cloud registration technique that employs unsupervised learning to reduce the dependence on labeled data. This approach enables the learning of rotation-sensitive features through rotation prediction. Drawing inspiration from the success of beam search in Natural Language Processing (NLP) [201], a hierarchical search method is proposed to search good initial rotation for cases with large rotations. This method combines a correspondence-free registration algorithm with a beam search scheme based on the octree data structure. By focusing on the most promising paths in the search space, this technique achieves efficient and memory-friendly point alignment. This chapter further provides a soft segmentation algorithm to solve the 3D partial point cloud registration built on the SoftClu introduced in the former Chapter (Chapter 3). It softly segments the point clouds into partitions in an unsupervised manner and then utilizes the IC-LK algorithm [202] to align the corresponding partitions. The differences between the proposed method and existing segmentation-based methods [203], [204] can be summed up in two points. First, the proposed algorithm segments the point cloud into geometrical partitions in an unsupervised way. Second, the proposed method is correspondence-free, but the existing segmentation-based point cloud registration methods belong to the correspondence-based approaches. To showcase the effectiveness of the suggested approach in both accuracy and time efficiency, thorough experiments were carried out on both synthetic and real datasets.

4.2 Methodology

This proposed approach builds upon PointNetLK [1], a technique for point cloud registration, and expands its capabilities to handle situations with large rotation and partial overlap. The approach utilizes PointNet to learn a point cloud embedding and a modified LK algorithm to estimate transformations accurately.

4.2.1 PointNet

This section begins with a brief overview of PointNet, a neural network used for processing 3D point clouds. A point cloud, denoted as $P = \{x_1, x_2, ..., x_N\} \subset \mathbb{R}^{3\times N}$, consists of N points,

each represented by a 3D coordinate $x_i \in \mathbb{R}^3$. PointNet is composed of two MLP layers and a max-pooling layer, represented by the function $\phi : \mathbb{R}^{3\times N} \to \mathbb{R}^K$. When applied to the point cloud P, $\phi(P)$ produces a K-dimensional vector descriptor, which captures important features of the point cloud. To prepare the point cloud for registration, the T-Net [79] component of the original PointNet has been removed.

4.2.2 Registration

Given two point clouds, template $\boldsymbol{P}^t = \{\boldsymbol{p}_i^t\}_{i=1}^N \in \mathbb{R}^{3\times N}$ and source $\boldsymbol{P}^s = \{\boldsymbol{p}_i^t\}_{i=1}^N \in \mathbb{R}^{3\times N}$, the primary objective of rigid registration is to determine the rigid-body transformation $G \in SE(3)$ that best aligns the source point cloud $\boldsymbol{P}^s$ with the template point cloud $\boldsymbol{P}^t$. The transformation G can be expressed using an exponential map as follows:

$$G = \exp\left([\xi]_\times\right), \tag{4.1}$$

where $\xi_i \in \mathbb{R},\ i = 1, 2, ..., 6,\ \xi = (\xi_1, \xi_2, ..., \xi_6)^\top \in \mathbb{R}^6$ that the first three parameters for the rotation (angle-axis representation) and last three jitter parameters for the translation, and $[\xi]_\times$ denotes the skew-symmetric matrix representation that translates the vector ξ to a matrix. For convenience, translation is denoted as $\boldsymbol{t} = (\xi_4, \xi_5, \xi_6)$ and angle-axis rotation representation is $\mathbf{r} = (\xi_1, \xi_2, \xi_3)$, with axis $\mathbf{r}/\|\mathbf{r}\|$ and angle $\|\mathbf{r}\|$. To make use of G, it augments the coordinates of each point in $\boldsymbol{P}^t$ and $\boldsymbol{P}^s$ with a 1, i.e., $(x, y, z)^\top \to (x, y, z, 1)^\top$. The problem of aligning 3D point clouds can be described as finding a transformation matrix G that satisfies the equation $\phi(\boldsymbol{P}^t) = \phi(G \cdot \boldsymbol{P}^s)$. The goal is to minimize the feature-metric projection error $Err_{G\in SE(3)}(G|\boldsymbol{P}^t, \boldsymbol{P}^s) = |\phi(G \cdot \boldsymbol{P}^s) - \phi(\boldsymbol{P}^t)|_2$ between the target point cloud $\boldsymbol{P}^t$ and the source point cloud transformed by G. The registration problem with initial transformation G_0 is then translated into the following optimization form:

$$F\left(\boldsymbol{P}^t, \boldsymbol{P}^s, G_0\right) \triangleq \min_{G\in SE(3)} Err(G|\boldsymbol{P}^t, G_0 \cdot \boldsymbol{P}^s). \tag{4.2}$$

Then, Levenberg-Marquardt (LM) [7] and inverse compositional (IC) [205] are applied to solve the optimization problem (4.2). Given an initial rotation vector $\mathbf{r}_0$ and translation $\boldsymbol{t}_0 = (0, 0, 0)$, the initial transformation matrix G_0 is calculated by Eq. (4.1), and the parameters of G is updated via the following rules:

$$\begin{aligned} &\boldsymbol{P}^s \leftarrow G_0 \cdot \boldsymbol{P}^s, J_i = \frac{\phi\left(\exp([-\Delta\xi]_\times) \cdot \boldsymbol{P}^t\right) - \phi(\boldsymbol{P}^t)}{\Delta\xi_i}, \\ &\xi = (J^\top J + \lambda I)^{-1} J^\top Err(G|\boldsymbol{P}^t, \boldsymbol{P}^s), \boldsymbol{P}^s \leftarrow \Delta G \cdot \boldsymbol{P}^s, \Delta G = \exp\left([\xi]_\times\right), \end{aligned} \tag{4.3}$$

where $\Delta\xi = (\Delta\xi_1, \Delta\xi_2, ..., \Delta\xi_6)^\top \in \mathbb{R}^6,\ \Delta\xi_i \in \mathbb{R},\ i = 1, 2, ..., 6$, and λ is the LM parameter that is utilized to ensure the numerical stability. According to [11], [205], the value $\Delta\xi_i$ is fixed to 2×10^{-2} on all iterations. Then, the final predicted $G_e = \Delta G_n \cdot ... \cdot \Delta G_1 \cdot \Delta G_0 \cdot G_0$ is the

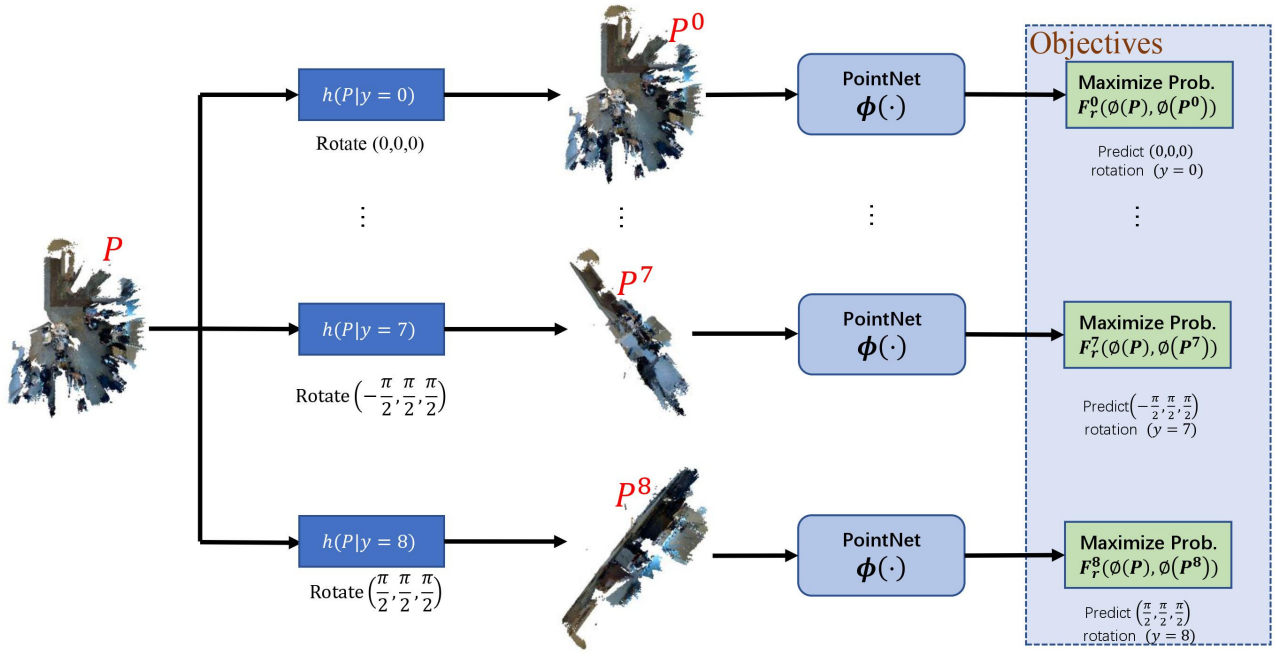


Figure 4.1: An overview of the proposed unsupervised feature learning framework.

composition of G_0 and all incremental estimations computed over n iterations (n is determined by a minimum threshold of ΔG. Through experience, the threshold is $\|\Delta G\| < 1e^{-7}$). The rotation matrix $\boldsymbol{R}_e$ and translation vector $\boldsymbol{t}_e$ can then be obtained from G_e, according to the following identity (4.4).

$$\begin{pmatrix} \boldsymbol{R}_e & \boldsymbol{t}_e^\top \\ \mathbf{0} & 1 \end{pmatrix} = G_e, \tag{4.4}$$

where $\mathbf{0} = (0, 0, 0)$. The objective of the registration loss function is to minimize the discrepancy between the estimated transformation, denoted as G_e, and the ground truth transformation, denoted as G_{gt}.

4.2.3 Rotation Based Unsupervised Feature Learning

The proposed unsupervised approach involves using an MLP model, denoted as $F_r(\cdot)$, to estimate the geometric transformation applied to a point cloud. As shown in Fig. 4.1, a set of $L \in \mathbb{N}$ discrete geometric transformations are defined as $H = \{h(\cdot|y)\}_{y=1}^{L}$, where $h(\cdot|y)$ is an operator that can be applied to a point cloud P to produce a transformed point cloud set $P^y = h(P|y)$, given the transformation label y. To estimate the unknown transformation label $y^\star$, $F_r(\cdot)$ takes as input the concatenation of feature embeddings $\phi(P)$ and $\phi(P^{y^\star})$. The model then outputs a probability distribution over all possible geometric transformations, allowing for selecting the most likely transformation.

$$F_r(P^{y^\star}|\theta) = \{F_r^y(\phi(P), \phi(P^{y^\star})|\theta)\}_{y=1}^{L}, \tag{4.5}$$

where $F_r^y(P^{y^\star})$ is the probability that the point cloud is transformed by transformation y. θ is the learnable parameter set of model F_r and ϕ. The training data is a set of M point cloud set, denoted as $D = \{X_i\}_{i=1}^M$. The unsupervised learning objective is:

$$\min_\theta \frac{1}{M} \sum_{i=1}^{M} loss(X_i, \theta), \tag{4.6}$$

with the loss function $loss(\cdot)$ satisfying:

$$loss(X_i, \theta) = -\frac{1}{L} \sum_{y=1}^{L} \log(F_r^y(\phi(X_i), \phi(h(X_i|y))|\theta)).$$

This algorithm focuses on defining a classification task through geometric transformations H, aimed at learning meaningful semantic features for tasks related to visual perception, such as point cloud registration or classification. The rotations included in the transformations are based on angle-axis vectors, specifically $[\pm\frac{\pi}{2}, \pm\frac{\pi}{2}, \pm\frac{\pi}{2}]$ and $[0, 0, 0]$, resulting in a total of $L = 9$ point cloud rotations. The registration model $\phi(\cdot)$ can be improved by using a registration loss function for fine-tuning.

4.2.4 Beam Search for Large Rotation Registration

Beam search owns a hyper-parameter k as beam size, an extension of greedy search in NLP tasks, such as machine translation and sentence generation, to effectively boost the sequential prediction performance. In the application scenario of point cloud registration, it applies the beam search scheme deals with the large rotation in 3D registration. However, two core problems will be solved: *i*) how to construct a rotation vector candidate set, and *ii*) how to effectively select k candidates in each step. Here introduce the solutions to the above two problems for registration.

Candidate Set Construction. The 3x3 rotation matrix $\boldsymbol{R}_\mathbf{r}$ in SO(3) can be obtained through the matrix exponential map from the angle-axis representation vector $\mathbf{r}$ as

$$\boldsymbol{R}_\mathbf{r} = \exp\left([\mathbf{r}]_\times\right) = \mathbf{I} + \frac{[\mathbf{r}]_\times \sin\|\mathbf{r}\|}{\mathbf{r}} + \frac{[\mathbf{r}]_\times^2 \left(1 - \cos\|\mathbf{r}\|\right)}{\mathbf{r}}. \tag{4.7}$$

By employing the angle-axis representation, it is possible to concisely depict the complete space of 3D rotations as a solid sphere of π radius in $\mathbb{R}^3$. Any rotation with an angle less than π can be uniquely represented by an angle-axis pair located inside the sphere, while rotations with angles equal to π have two possible representations on the surface of the ball. For simplicity, the minimal cube with dimensions $[-\pi, \pi]^3$ is utilized as the rotation domain, which encloses the π-ball. The beam search process involves dividing the initial cubes into smaller sub-cubes using an octree data structure. This process is repeated multiple times until convergence is achieved.

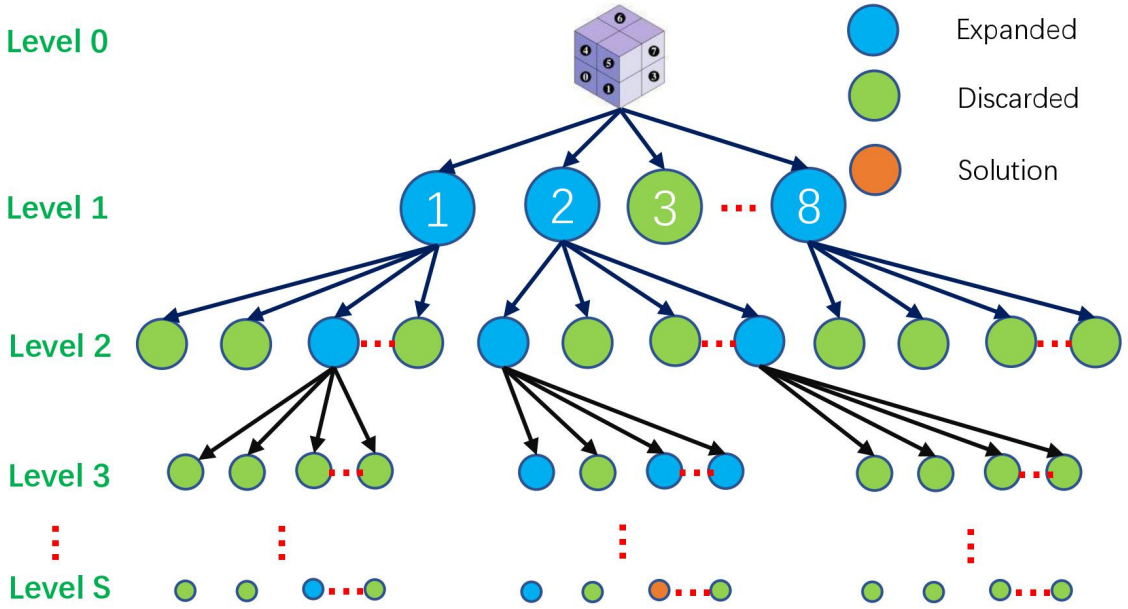


Figure 4.2: An overview of the proposed beam search process.

Candidates Selection. At each level, it applies the center point of each sub-cube in the candidate set as the initial vector $\mathbf{r}_0$ and $\boldsymbol{t}_0 = (0, 0, 0)$ to obtain G_0. G_0 is then taken into Eq.(4.4) to get $\boldsymbol{R}_e$ and $\boldsymbol{t}_e$. Next, k sub-cubes are selected with the minimal modified Chamfer distance loss between $\boldsymbol{P}^t$ and transformed $\boldsymbol{P}^s$ (by $\boldsymbol{R}_e$ and $\boldsymbol{t}_e$) from the candidate set. The loss function is defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{R}_e, \boldsymbol{t}_e, \boldsymbol{P}^s, \boldsymbol{P}^t) = &\frac{1}{|\boldsymbol{P}^s|}\sum_{p\in\boldsymbol{P}^s}\rho\left(\min_{q\in\boldsymbol{P}^t}\|\boldsymbol{R}_e p + \boldsymbol{t}_e - q\|_2\right) \\ &+ \frac{1}{|\boldsymbol{P}^t|}\sum_{q\in\boldsymbol{P}^t}\rho\left(\min_{p\in\boldsymbol{P}^s}\|\boldsymbol{R}_e p + \boldsymbol{t}_e - q\|_2\right),\end{aligned} \tag{4.8}$$

where $\rho(x) = \min(\sigma, x)$ is with the threshold σ for clipping the distance. The flowchart of the beam search algorithm for alignment is illustrated in Figure 4.2.

The computation is summarized in Algorithm 3. In lines 8 and 25, a list (cube_list) is used to store all sub-cubes divided from the current optimal k cubes. The function of $subdivide(\cdot)$ subdivides each cube stored in cube_list into eight sub-cubes (octants) and stores all octants into a new list. As for the stop criterion, once the Chamfer distance loss for the current cube is less than a threshold ϵ or the number of loops achieves the maximal iteration S, the beam search stops. By doing this, the joint framework seamlessly incorporates both the global beam search and the local learning-based search.

Algorithm 3 Beam Search for Registration(Python syntax).

Input: $\boldsymbol{P}^s, \boldsymbol{P}^t$, beam size k, iterations S and threshold ϵ.

Output: G_e

$\xi \leftarrow (0, 0, ..., 0)^\top$
$G_e \leftarrow F\left(\boldsymbol{P}^s, \boldsymbol{P}^t, \exp\left([\xi]_\times\right)\right)$
$\boldsymbol{R}, \boldsymbol{t} \leftarrow G_e[0:3, 0:3], G_e[0:3, 3]$
$min_{dis} \leftarrow \mathcal{L}(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{P}^s, \boldsymbol{P}^t)$
if $min_{dis} < \epsilon$ **then**
    return G_e
end if
cubes $\leftarrow$ *subdivide*([π-ball]) # [π-ball] represents a list with only one element π-ball, and cubes is also a list.
dis_list $\leftarrow$ *list*()
for i in *range*$(0, S)$ **do**
    for sub_cube in cubes **do**
        $\xi_1, \xi_2, \xi_3 \leftarrow$ *center*(sub_cube) # *center*$(\cdot)$ gets the center coordinate of a sub_cube.
        $\xi \leftarrow (\xi_1, \xi_2, \xi_3, 0, 0, 0)^\top$
        $G_e \leftarrow F\left(\boldsymbol{P}^s, \boldsymbol{P}^t, \exp\left([\xi]_\times\right)\right)$
        $\boldsymbol{R}, \boldsymbol{t} \leftarrow G_e[0:3, 0:3], G_e[0:3, 3]$
        $dis \leftarrow \mathcal{L}(\boldsymbol{R}, \boldsymbol{t}, \boldsymbol{P}^s, \boldsymbol{P}^t)$
        dis_list $\leftarrow$ dis_list.append(dis)
        if $dis < \epsilon$ **then**
            return G_e
        end if
        if $min_{dis} < dis$ **then**
            $min_{dis} \leftarrow dis, G_e \leftarrow G$
        end if
        top_cubes $\leftarrow$ *topK*(dis_list, cubes, k)
        cubes $\leftarrow$ *subdivide*(top_cubes)
    end for
end for
return G_e

4.2.5 Partial Point Cloud Registration

An overview of partially overlapping registration approach is shown in Fig. 4.3, point clouds $\boldsymbol{\mathcal{P}}^s$ and $\boldsymbol{\mathcal{P}}^t$ are passed through a shared PointNet [79], ϕ, to compute point-wise descriptors $\boldsymbol{\mathcal{F}}^s = \phi(\boldsymbol{\mathcal{P}}^s) \in \mathbb{R}^{d \times N}$ and $\boldsymbol{\mathcal{F}}^t = \phi(\boldsymbol{\mathcal{P}}^t) \in \mathbb{R}^{d \times N}$. Then, a shared soft segmentation module g acts

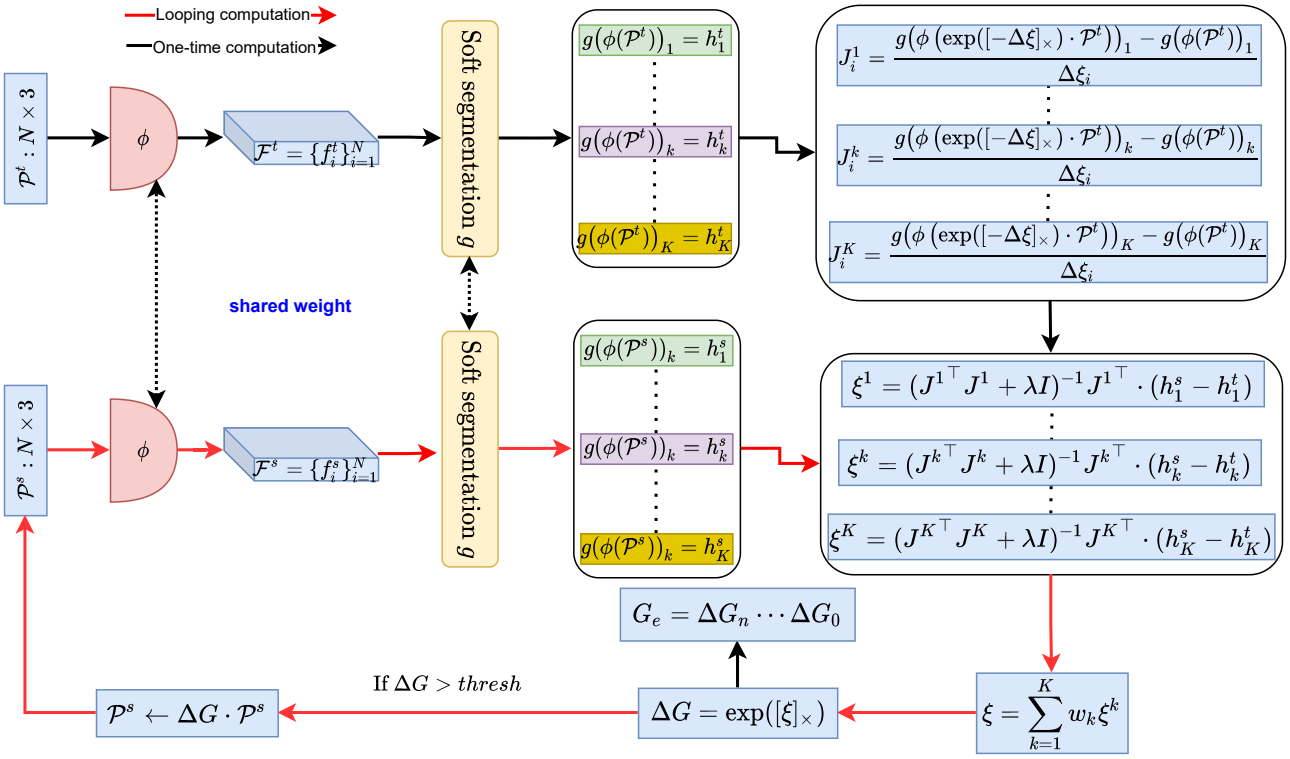


Figure 4.3: **Overview of the proposed registration framework for partial overlaps.**

on $\boldsymbol{\mathcal{F}}^s$ and $\boldsymbol{\mathcal{F}}^t$ to produce K partition centers for each point cloud. It first denotes partition centers as $[\boldsymbol{h}_1^s, \cdots, \boldsymbol{h}_K^s] = g(\boldsymbol{\mathcal{F}}^s) \in \mathbb{R}^{d\times K}$ and $[\boldsymbol{h}_1^t, \cdots, \boldsymbol{h}_K^t] = g(\boldsymbol{\mathcal{F}}^t) \in \mathbb{R}^{d\times K}$ for $\boldsymbol{\mathcal{P}}^s$ and $\boldsymbol{\mathcal{P}}^t$, respectively. The Jacobian J^k of the partition k is computed once using $g\left(\phi\left(\boldsymbol{\mathcal{P}}^t\right)\right)_k$ with $k = 1, 2, \cdots, K$. The optimal twist parameters $\xi^k \in \mathbb{R}^6$ for each partition k are then computed and the final twist parameter ξ is obtained by taking the weighted sum of all partitions, $\xi = \sum_{k=1}^K w_k \xi_k$. The final twist parameter is utilized to gradually modify the transformation of $\boldsymbol{\mathcal{P}}^s$, and the center vector $\boldsymbol{h}_k^s$ is recalculated. The w_k is the weight coefficient. The final predicted transformation $G_e^j = \Delta G_n \cdots \Delta G_1 \cdot \Delta G_0 \cdot G_0$ is the composition of all incremental estimates $\Delta G_i, i = 0, 1, \cdots, n$, computed with n iterations and with initial transformation G_0. Finally, the partially overlapping registration can be summarized into two critical steps: soft segmentation and soft segmentation-based registration. Next will introduce them in detail.

4.2.6 Soft segmentation

The soft segmentation, built upon SoftClu discussed in Chapter 3, but does not consider the prototypes in the feature space. It assigns each point $\boldsymbol{p}_i^l \in \boldsymbol{\mathcal{P}}^l$ to one of K potential categories or geometric partitions, denoted by the superscripts $l = s, t$. Specifically, feature map $\boldsymbol{\mathcal{F}}^l$ are processed by a classification head g that outputs a class probability matrix $\boldsymbol{S}^l = \{s_{ij}^l \in [0, 1]\}_{i,j}^{N,K}$. Three fully connected layers form g. Each layer is composed of a linear layer followed by batch normalization. With the exception of the last layer, each layer has a LeakyReLU

activation function. The final layer outputs N vectors with K dimensions, which is equivalent to the number of segmentation categories. The soft centers or prototypes of the partition j in both geometric space are computed as

$$\boldsymbol{c}_j^l = \frac{1}{\sum_{i=1}^N s_{ij}^l} \sum_{i=1}^N s_{ij}^l \boldsymbol{p}_i^l \tag{4.9}$$

where $j = 1, 2, \cdots, K$. $\boldsymbol{S}^l$ denotes the allocation of each point in $\boldsymbol{\mathcal{P}}^l$ to K distinct spatial partitions, expressed as a probabilistic distribution. Thus, if $\boldsymbol{p}_i^l$ belongs to partition $j^\star$, point $\boldsymbol{p}_i^l$ and prototype $\boldsymbol{c}_{j^\star}^l$ should have the shortest distance among the distances of $\boldsymbol{p}_i^l$ with other prototypes, i.e., $\|\boldsymbol{p}_i^l - \boldsymbol{c}_{j^\star}^l\|_2 \leq \|\boldsymbol{p}_i^l - \boldsymbol{c}_j^l\|_2, j \neq j^\star$. This constraint can be translated into minimizing the average cross-entropy loss between $\boldsymbol{\gamma}^l$ and $\boldsymbol{S}^l$, i.e.,

$$\begin{aligned} \boldsymbol{\gamma}^l &= \{\gamma_{ij}^l = \text{softmax}\left(\alpha - \|\boldsymbol{p}_i^l - \boldsymbol{c}_j^l\|\right)\}_{i,j}^{N,J} \\ \mathcal{E}(\boldsymbol{\gamma}^l, \boldsymbol{S}^l) &= -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^J \boldsymbol{\gamma}_{ij}^l \log s_{ij}^l. \end{aligned} \tag{4.10}$$

Here $\alpha > 0$ is a learning parameter. To avoid the trivial solution that most points are assigned to the same cluster, it further introduces the entropy of cluster assignment probabilities

$$\mathcal{H}\left(\boldsymbol{S}^l\right) = -\sum_{j=1}^J P\left(\boldsymbol{s}_j^l\right) \log P\left(\boldsymbol{s}_j^l\right), \tag{4.11}$$

where $P\left(\boldsymbol{s}_i^l\right) = \frac{\sum_{i=1}^N s_{ij}^l}{\|\boldsymbol{S}^l\|_1}, l \in \{s, t\}$ is the assignment probability of partition i. The final soft segmentation loss is

$$\mathcal{L}_{seg} = \sum_l \mathcal{E}(\boldsymbol{\gamma}^l, \boldsymbol{S}^l) + \mathcal{H}\left(\boldsymbol{S}^l\right). \tag{4.12}$$

4.2.7 Soft segmentation-based registration

The soft segmentation-based registration uses the above segmentation results to estimate the transformation without searching correspondences. Specifically, for ideal consistent point clouds, the registration problem can then be described as searching G satisfying $g(\phi(\boldsymbol{\mathcal{P}}^t))_k = g(\phi(G \cdot \boldsymbol{\mathcal{P}}^s))_k$. The registration problem can be translated into minimizing the feature-metric projection error between $\boldsymbol{\mathcal{P}}^t$ and transformed $\boldsymbol{\mathcal{P}}^s$ related to the soft partition $k (k = 1, 2, \cdots, K)$.

$$\min_{G \in SE(3)} \|g(\phi(\boldsymbol{\mathcal{P}}^t))_k - g(\phi(G \cdot \boldsymbol{\mathcal{P}}^s))_k\|_2. \tag{4.13}$$

However, input point clouds are usually not consistent in practical applications due to partial overlapping. Considering that if $\boldsymbol{p}_i \to \boldsymbol{q}_j$ and $\boldsymbol{p}_k \to \boldsymbol{q}_l$ are correct correspondences, then the distance between $\boldsymbol{p}_i$ and $\boldsymbol{p}_k$ should be similar to the distance between $\boldsymbol{q}_j$ and $\boldsymbol{q}_l$. Based on this intuition, a quadratic constraint is thus introduced to control the contribution (weight) of each

partition. Specifically, here first defines two matrices $A^l = \{\|\boldsymbol{c}_i^l - \boldsymbol{c}_j^l\|_2\}_{i,j}^{K,K}, l = s, t$ and a vector $\boldsymbol{v} = -|A^s - A^t| \cdot \mathbf{1}_K$. IC-LK algorithm [205] is then used to solve the optimization problem in Eq. (4.13). Incorporating with the quadratic constraint, the parameters of G are updated by the following rules.

$$
\begin{aligned}
J_i^k &= \frac{g\left(\phi\left(\exp([-\Delta\xi]_\times) \cdot \boldsymbol{\mathcal{P}}^t\right)\right)_k - g\left(\phi(\boldsymbol{\mathcal{P}}^t)\right)_k}{\Delta\xi_i}, \\
\xi_k &= (J^{k\top} J^k + \lambda I)^{-1} J^{k\top} \|g(\phi(\boldsymbol{\mathcal{P}}^t))_k - g(\phi(G \cdot \boldsymbol{\mathcal{P}}^s))_k\|_2, \\
\boldsymbol{v} &= -|A^s - A^t| \cdot \mathbf{1}_K, w_k = \text{softmax}(\boldsymbol{v})_k, \xi = \sum_{k=1}^{K} w_k \xi_k, \\
\boldsymbol{\mathcal{P}}^s &\leftarrow \Delta G \cdot \boldsymbol{\mathcal{P}}^s, \Delta G = \exp\left([\xi]_\times\right),
\end{aligned}
$$

where $\Delta\xi = (\Delta\xi_1, \Delta\xi_2, ..., \Delta\xi_6)^\top \in \mathbb{R}^6$, $\Delta\xi_i \in \mathbb{R}$, $i = 1, 2, ..., 6$, and $\lambda = 0.001$ is a Levenberg-Marquardt [7] parameter utilized to ensure the numerical stability. The final predicted $G_e = \Delta G_n \cdot ... \cdot \Delta G_1 \cdot \Delta G_0$ is the composition of all incremental estimates computed with n iterations (n is decided by a minimum threshold for ΔG.

4.3 Experiments

4.3.1 Implementation details and evaluation metrics

All implementations are built on the Pytorch [206] library. All of the proposed models were trained on two Tesla V100-PCI-E-32G GPUs. The proposed registration algorithm is evaluated using three metrics: Clip Chamfer Distance (CD), Rotation Error (RE), and Translation Error (TE). The success rate (RR) of the alignment is measured by the percentage of cases where the rotation and translation errors are below set thresholds. The Rotation Error (RE) is defined as $RE = \arccos \frac{\mathbf{Tr}\left(\boldsymbol{R}^\top \boldsymbol{R}^\star\right) - 1}{2}$ and the Translation Error (TE) is defined as $TE = |\boldsymbol{t} - \boldsymbol{t}^\star|_2$, where $\boldsymbol{R}^\star$ and $\boldsymbol{t}^\star$ represent the ground-truth rotation matrix and translation vector, respectively. The Clip Chamfer Distance measures the proximity of the two point clouds to each other and is calculated as follows:

$$
CD(\hat{\boldsymbol{P}}^s, \boldsymbol{P}^t) = \sum_{\boldsymbol{p} \in \hat{\boldsymbol{P}}^s} \rho\left(\min_{\boldsymbol{q} \in \boldsymbol{P}^t} \|\boldsymbol{R}_e \boldsymbol{p} + \boldsymbol{t}_e - \boldsymbol{q}\|_2, \sigma\right) + \sum_{\boldsymbol{q} \in \boldsymbol{P}^t} \rho\left(\min_{\boldsymbol{p} \in \hat{\mathcal{P}}^s} \|\boldsymbol{R}_e \boldsymbol{p} + \boldsymbol{t}_e - \boldsymbol{q}\|_2, \sigma\right), \quad (4.14)
$$

where $\rho(x, \sigma) = \min(\sigma, x)$ is with the threshold $\sigma > 0$ for clipping the distance. After registration, $\hat{\boldsymbol{P}}^s$ is utilized to represent the source point cloud that has been transformed. The parameters of the MLP in ϕ are set as $[3, 64, 64, 64, 128, K = 1024]$ and the MLP for F_r are set as $[2048, 1024, 512, 256, 9]$. All models were trained using AdamW optimizer [193] with a base learning rate of 0.001. PointNetLK with unsupervised learning is denoted as **ROT**, PointNetLK with Beam Search as **PNLKBS**, and the full model combining beam search with ROT as **ROTBS**. The model for partial overlaps is signed as **PROTBS**.

4.3.2 Datasets

ModelNet40 [207] is a dataset consisting of 12,311 CAD models from 40 categories. To assess the generality of various models, the dataset has been divided into two parts, with 20 categories for training and 20 categories for cross-category testing. In the same-category experiments, the dataset is split into an 80/20 ratio for training and testing, respectively. Each source point cloud is sampled from ModelNet40 with 1024 points, normalized into a unit box at the origin $[0, 1]^3$. Rigid transformations along each axis are generated to create the target points, with rotations sampled in $[0, 45°]$ and translations in $[-0.5, 0.5]$. To test the ability of large rotation registration, misalignment translations and rotations for testing are in the range of $[0, 0.5]$ and $[0, 180°)$, respectively. Gaussian noise is added to the point clouds to test the robustness of the proposed registration methods, with Gaussian noise sampled from $\mathcal{N}(0.0, 0.01^2)$ and clipped to $[-0.05, 0.05]$. Partial-to-partial registration, the most challenging case for point cloud registration, is also tested, following the protocol in [133] to generate partial-to-partial point cloud pairs closer to real-world applications, by creating a half-space with a random direction for each point cloud and shifting it to retain approximately 70% of the points.

7Scene [208] consists of RGB-D camera frames that have been tracked across seven different scenes: Chess, Fires, Heads, Office, Pumpkin, Redkitchen, and Stairs. The dataset is split into two parts, with 296 scans for training and 57 scans for testing. The point clouds in the dataset have been uniformly sampled twice, with one of the samples being subjected to a rigid transformation to simulate differences in pose. To generate the target points, a random rigid transformation is applied along each axis, with rotation being sampled within the interval $[0, 45°]$ and translation being sampled within the interval $[-0.5, 0.5]$. Each point cloud in the dataset consists of 2048 randomly sampled points.

3DMatch [19] is a collection of 3D point cloud pairs from various real-world scenes that includes ground truth transformations [12]. This dataset contains realistic registration challenges such as partial overlap, noise, and outliers. The experiment follows the standard train/test split procedure and generates pairs with at least 30% overlap for training and testing purposes [12], [119]. The point clouds are downsampled using 5cm voxels and randomly subsampled to generate point clouds with uniform density. Each point cloud finally consists of 5000 points that are randomly selected from the downsampled point clouds.

4.3.3 Baseline

This section showcases the superiority of the proposed method by comparing its performance with that of traditional optimization-based methods, recent correspondence-learning methods,

and correspondence-free methods.

- Non-learning-based methods. For comparison with the traditional registration method, the Fast Global Registration (FGR) [75] and the Optimal Solution to 3D ICP (Go-ICP) [209] have been selected as the comparison methods. FGR is a commonly used classic registration approach.

- Correspondence-learning methods. For comparison with correspondence-learning techniques, the state-of-the-art RPMNet [133], Deep Closest Point (DCP) [10], the latest Robust Graph Matching (RGM) [15], and Deep Global Registration (DGR) [12] are selected as comparison methods.

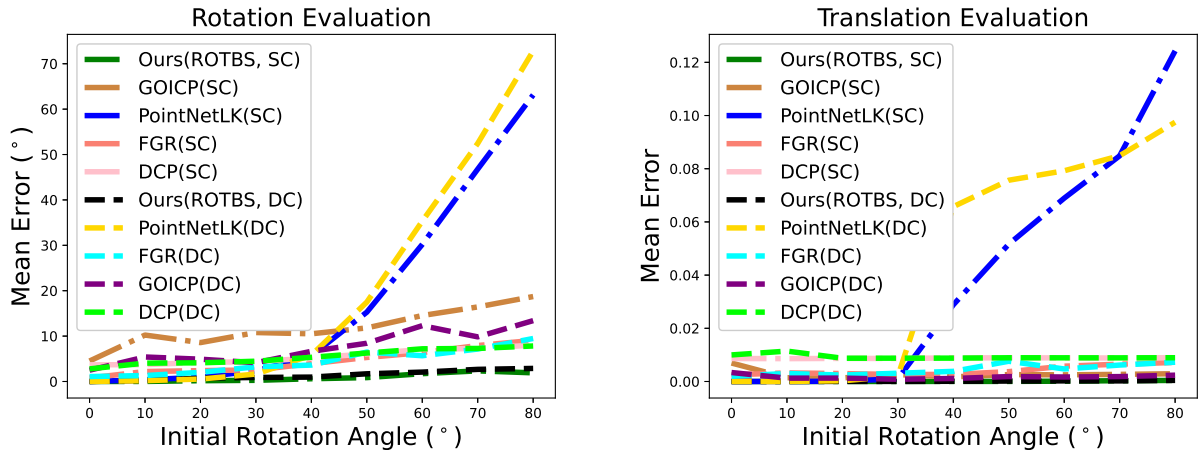
- Correspondences-free methods: To compare with the correspondences-free methods, PointNetLK [1], FMR [11], and the latest PointNetLK revisits (PNetLKR) [210] are selected as baselines.

PointNetLK and FMR are initialized with an identity transformation matrix. For DCP, RPMNet, PointNetLK, FMR, PNetLKR, RGM, and DGR, the code was partially adapted from the authors' release. The cython version py-goicp 0.0.4 of the Go-ICP project was used for Go-ICP. The implementations of FGR were taken from Intel Open3D.

4.3.4 Evaluation on ModelNet40

The following section evaluates the performance of the registration when PointNet is trained through an unsupervised approach without the use of any category labels on ModelNet40. Then, the section examines the registration performance when features are trained with unsupervised learning as additional information in a standard classification task. The hyperparameters of Algorithm (3) such as λ, k, ϵ and S are set to 0.0, 8, $1e$-4 and 5 respectively. Finally, the section investigates partial overlapping registration, where K is set to 64 unless specified otherwise. This is because similar performance can be achieved with $32 \leq K \leq 128$ when the overlap ratio is 70% on ModelNet40 [207]. The models are trained for 200 epochs for PointNet and 300 epochs for registration with a batch size of 32 and a learning rate that decreases by 0.7 every 20 epochs. The experiments are conducted on 20 object categories for training and testing on either the same categories (SC) or different categories (DC).

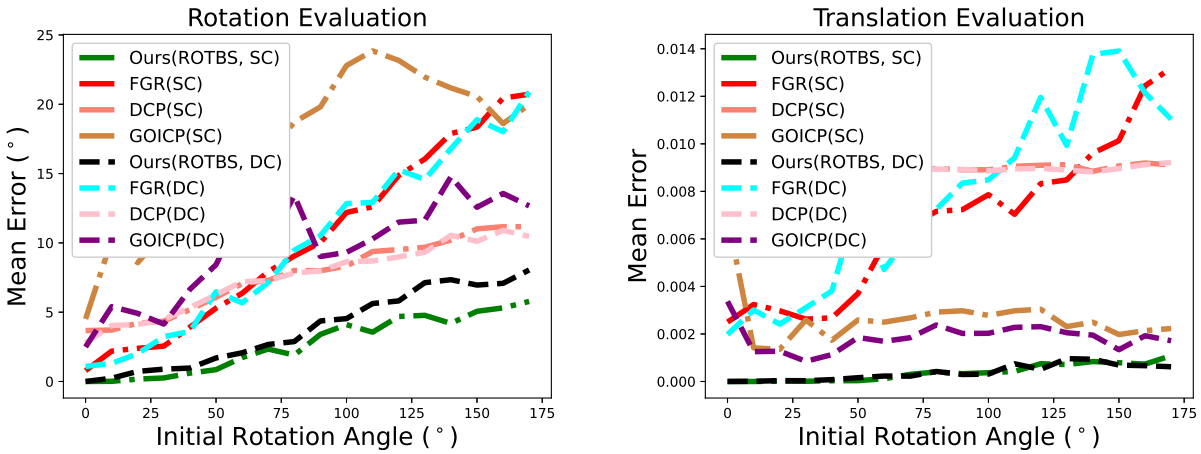
Unsupervised learning for point cloud registration. The objective of the experiment is to evaluate the performance of the registration task using PointNet with a complete unsupervised approach. The proposed method was compared with four methods: FGR, PointNetLK, Go-ICP, and DCP. The comparison results on the ModelNet40 dataset with random translations and different initial rotations are displayed in Figure 4.4. The results showed that the proposed



(a) (b)

Figure 4.4: The comparison of the results obtained from the registration process on ModelNet40 is presented. The vertical axis represents the error in rotation (a) and translation (b) after the registration process, while the horizontal axis depicts the initial misalignment between the template and source point clouds.

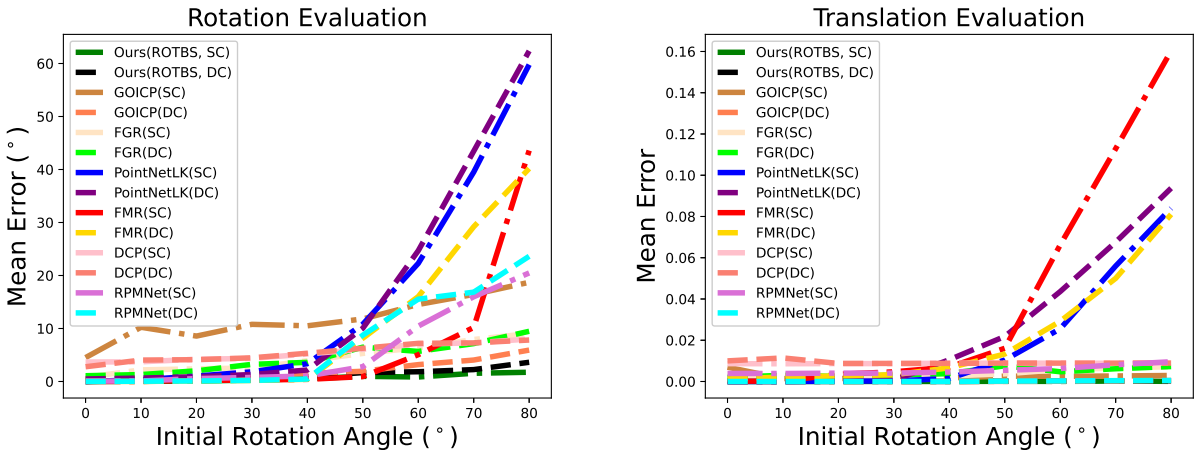
unsupervised method outperformed the other methods, FGR, Go-ICP, and PointNetLK, due to two main factors. Firstly, the rotation-sensitive features extracted from the point cloud improved the performance of the registration, since the registration process requires the global features to pose awareness. Secondly, the beam search method helped to provide an appropriate initial rotation for alignment, thus handling large rotations. When comparing the results more closely, it was observed that the absolute mean errors of PointNetLK exceeded 10° when the rotation exceeded 50°, rendering the results meaningless. On the other hand, the proposed method, ROTBS, consistently achieved accurate results with absolute errors less than 7° due to the beam search. Furthermore, ROTBS outperformed the supervised method, DCP, overall and the difference became more pronounced when the initial misalignment was greater than 60°. Even when the rotation angle ranged from 50° to 170°, ROTBS still maintained perfect results with absolute errors less than 10°, as shown in Figure 4.5. The results of ROTBS were compared with global methods (FGR, Go-ICP, and DCP) in Figure 4.5. To test the generalizability of the proposed method, Figures 4.4 and 4.5 also presented the comparison of results when training and testing on different object categories on ModelNet40. ROTBS still obtained accurate alignment on unseen object categories during the training process. In conclusion, the experimental results indicate that the proposed registration framework has broad application prospects, as it does not require any labels to train the encoder and produces a global solution.



(a) (b)

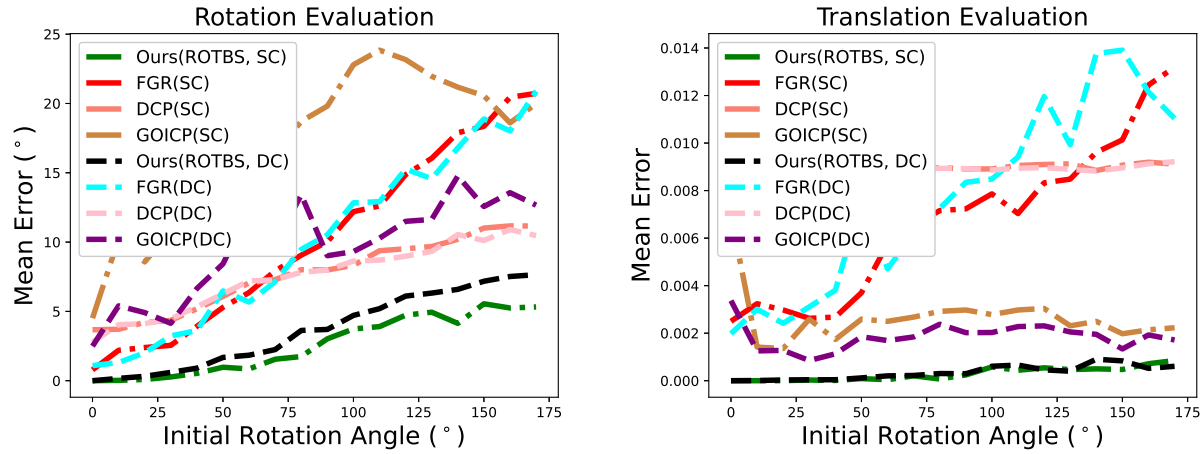
Figure 4.5: Comparison of the large rotation registration results with DCP, GO-ICP and FGR. The vertical axis shows the rotation (a) and translation (b) error, and the horizontal axis shows the initial misalignment between the template and source.

Unsupervised learning as enhancement information for registration. The PointNet is a deep learning architecture that has been trained using unsupervised learning for the purpose of enhancing its performance in a standard classification task. A comparison study was conducted between the proposed ROTBS (Rotation Beam Search) method and other state-of-the-art methods, such as DCP (Dense Correspondence Prediction) and RPMNet (Rotation Prediction and Matching Network), as shown in Figure 4.6. The comparison was performed by randomly translating the point clouds and by introducing different initial misalignment angles between the template and source point clouds. The results indicate that ROTBS outperforms the other methods in terms of accuracy in alignment, regardless of whether the categories in the testing dataset were visible or not during the training phase. This highlights the impact of the enhancement information in making the feature extractor more sensitive to rotation and how beam search can aid local learning-based methods in finding the most appropriate transformations, thereby improving the overall registration quality. When the initial misalignment between the template and source point clouds ranges from 0° to 180°, as shown in Figure 4.7, the results of the comparison between the proposed ROTBS method and the other methods were still favorable for ROTBS. The proposed method was able to achieve mean rotation errors of less than 7.5 degrees and translation errors of less than 0.001, even for object categories that were not seen during the training phase. This further supports the effectiveness of ROTBS in solving large rotation alignment tasks and demonstrates its superiority over other existing methods in this field.



(a) (b)

Figure 4.6: A comparison of the results of the registration process where the unsupervised method improves the information is presented. The vertical axis illustrates the errors in rotation (a) and translation (b) after the registration process, while the horizontal axis displays the initial misalignment between the template and the source.



(a) (b)

Figure 4.7: Comparison of the large rotation registration results with FGR, Go-ICP and DCP, where the unsupervised approach enhances the information on Model. The vertical axis shows the rotation (a) and translation (b) error, and the horizontal axis shows the initial misalignment between the template and source.

Partially overlapping unseen object. The results of the study are displayed in the left column of Table 4.1 (Unseen objects). The proposed method, PROTBS, demonstrates the best performance and significantly surpasses both traditional methods and the strongest learning-based methods. When comparing with the correspondence-free methods PointNetLK and

PNetLKR, the proposed approach shows a substantial improvement in all evaluation metrics. It is noteworthy that the proposed approach performs similarly to RGM, which is a state-of-the-art method trained in a supervised manner. This highlights the effectiveness of the soft segmentation-based registration, as it allows the focus to be placed on the overlap regions, thereby improving the overall performance of the registration process.

Table 4.1: Results of the comparison on a partially overlapping ModelNet40 dataset with Gaussian noise are presented below. The results are highlighted in bold font, indicating the best performance achieved among all methods evaluated.

Method	Unseen objects			Unseen categories		
	CD	RE	TE	CD	RE	TE
ICP [42]	0.115	24.88	0.267	0.119	26.64	0.278
FGR [75]	0.121	42.43	0.302	0.124	41.96	0.291
RPMNet [133]	0.085	1.70	0.018	0.087	1.98	0.023
RGM [15]	0.082	0.93	0.**009**	0.085	1.55	0.014
PointNetLK [1]	0.124	29.72	0.291	0.161	32.69	0.312
PNetLKR [210]	0.108	24.35	0.194	0.142	25.56	0.250
PROTBS	**0.082**	**0.89**	0.010	**0.083**	**1.17**	**0.014**

Partially overlapping unseen category. This experiment assesses the the capacity of various models to generalize in the context of learning-based point cloud registration for previously unseen categories that were not part of the training data. The results, listed in the right column of Table 4.1 (Unseen categories), demonstrate that PROTBS outperforms other models in this aspect. It is also evident that the performance of traditional methods remains relatively unchanged. Although RGM performs well in generalization, the proposed method outperforms it, as it is an unsupervised method and does not rely on any category labels. However, it is clear that the other learning-based methods struggle to generalize well to unseen categories.

Figure 4.8 shows the visualizations of registration. These results further demonstrate the effectiveness of incorporating soft segmentation into the registration process.

4.3.5 Evaluation on 7Scene

Further experiments were conducted on the real-world 7Scene dataset [208] to evaluate the proposed method. The model was trained in an unsupervised manner with a batch size of 32 and the number of clusters, K, was set to 72 for the clustering module. The results, as shown in Table 4.2, demonstrate the competitiveness of the proposed method with a 1.66 rotation error, 0.007 translation error, and 0.041 clip chamfer distance error. Compared to correspondence-free

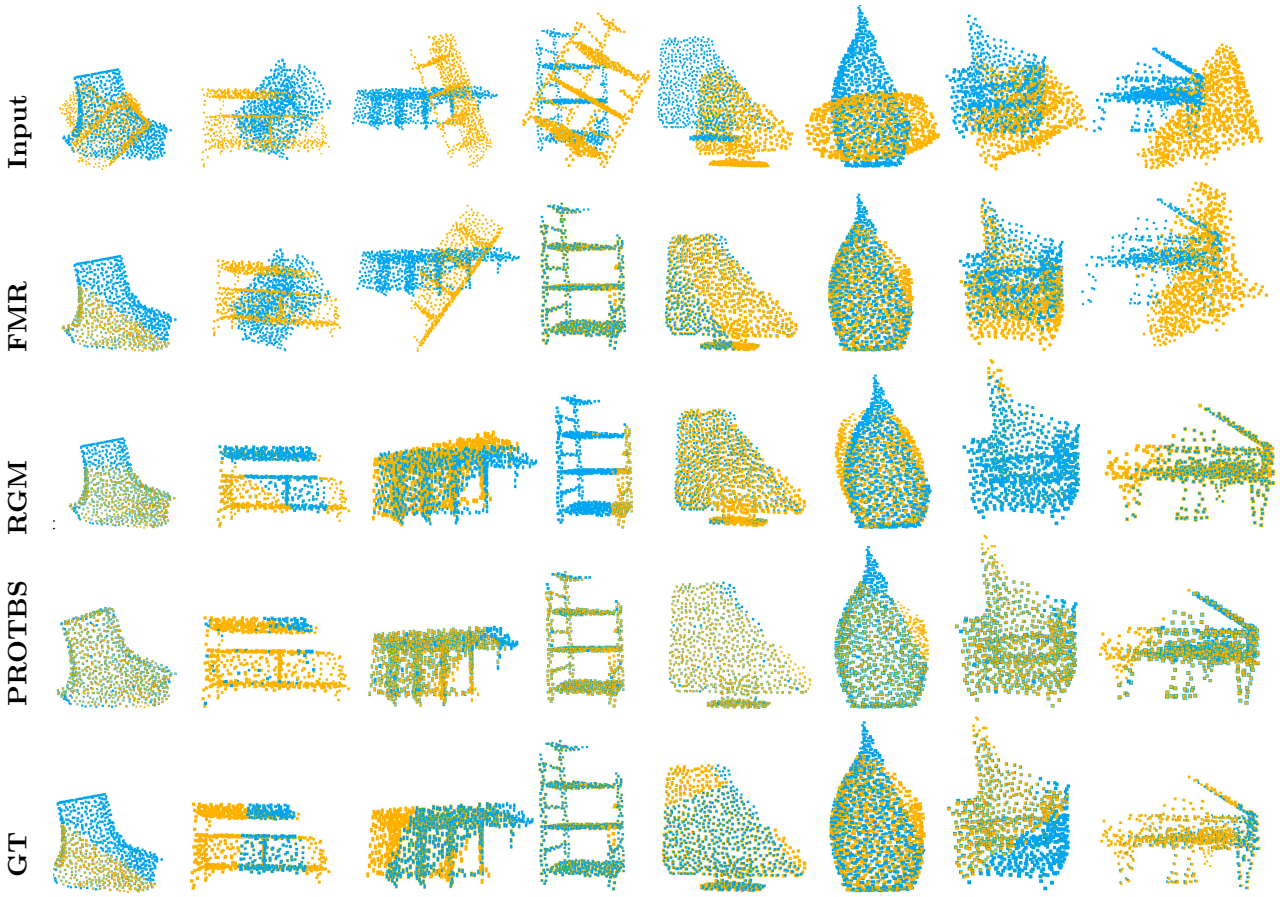


Figure 4.8: Qualitative registration examples on partially overlapping ModelNet40 dataset

methods such as PointNetLK, PNetLKR, and FMR, as well as correspondence-based methods like RGM and non-learning-based methods like ICP and FGR, the proposed method outperforms all of them in all evaluation criteria. This shows that the PROTBS method can be applied to more complex scenes with great success.

Table 4.2: Registration results of the comparison on the 7Scene dataset. The results are highlighted in bold font, indicating the best performance achieved among all methods evaluated.

Method	Year	CD	RE	TE
ICP [42]	1992	0.118	7.69	0.121
FGR [75]	2016	0.051	1.92	0.009
RGM [15]	2021	0.044	1.79	0.008
PointNetLK [1]	2019	0.067	3.94	0.043
FMR [11]	2020	0.063	3.66	0.012
PNetLKR [210]	2021	0.058	2.67	0.018
PROTBS	-	**0.041**	**1.66**	**0.007**

4.3.6 Evaluation on 3DMatch

Since no category labels are available in the 3DMatch dataset, the proposed PROTBS is trained directly in an unsupervised manner. The value of K is still set to 64. The experiment involves training PointNet for 250 epochs and the registration process for 400 epochs. The evaluation metric used is the Recall, which is calculated as the percentage of point cloud pairs with a rotation error (30cm) and translation error below a threshold (15 degrees) compared to the total number of pairs. The performance of PROTBS is compared to other methods such as PointNetLK, DCP, DGR, FGR, and Go-ICP. Table 4.3 summarizes the results of the performance comparison on the 3DMatch benchmark, and it is shown that PROTBS performs well and outperforms other methods. Specifically, it achieves an 89.14% Registration Recall, which surpasses the second-best DGR method with 85.2%.

A visual example of PROTBS is presented in Figure 4.9, demonstrating its ability to achieve accurate results in challenging indoor scenes with a low overlap ratio.

Table 4.3: Registration results on 3DMatch.

Method	Recall	TE	RE(°)
Go-ICP	22.9%	14.7	5.38
FGR	42.7%	10.6	4.08
PointNetLK	1.61%	21.3	8.04
FMR	14.3%	13.0	4.33
DCP	3.22%	21.4	8.42
DGR	85.2%	7.73	2.58
PROTBS (ours)	**89.14%**	**6.54**	**2.45**

4.3.7 Ablation study

The ablation studies were performed on the ModelNet40 dataset to evaluate the significance of each component of our proposed method, referred to as ROTBS. The features were extracted using an unsupervised method, and the results, shown in Figure 4.10, demonstrate that the unsupervised approach can enhance the performance of PointNetLK. Although the improvement is marginal compared to PNLKBS, using an unsupervised loss can lead to more excellent numerical stability. Additionally, the beam search strategy effectively improves the alignment accuracy, especially for registration tasks that involve large rotations.

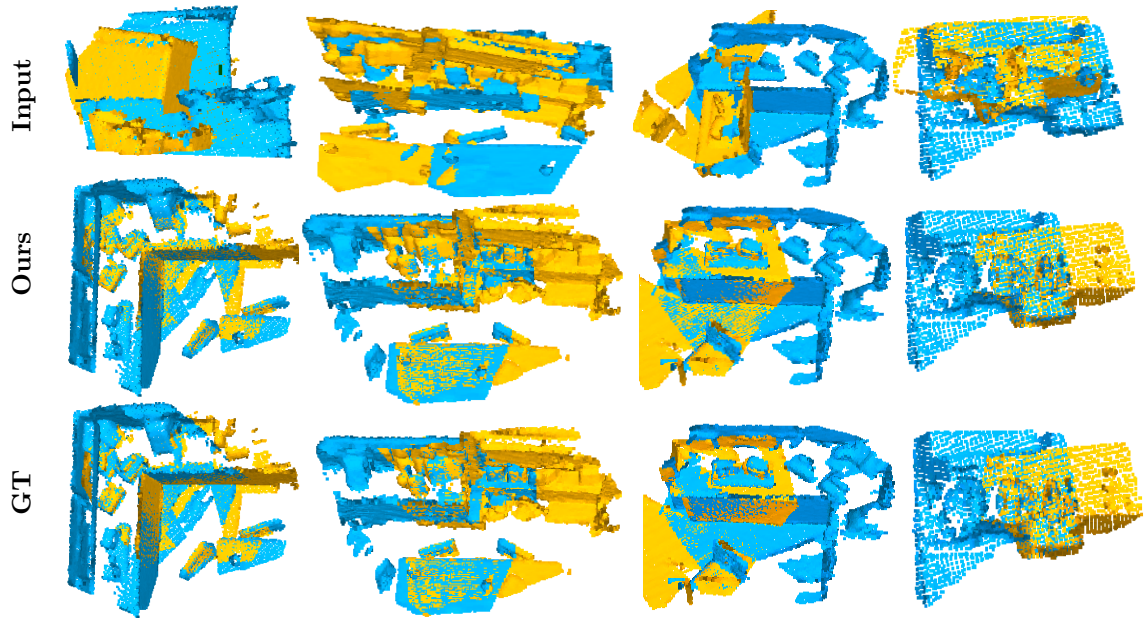
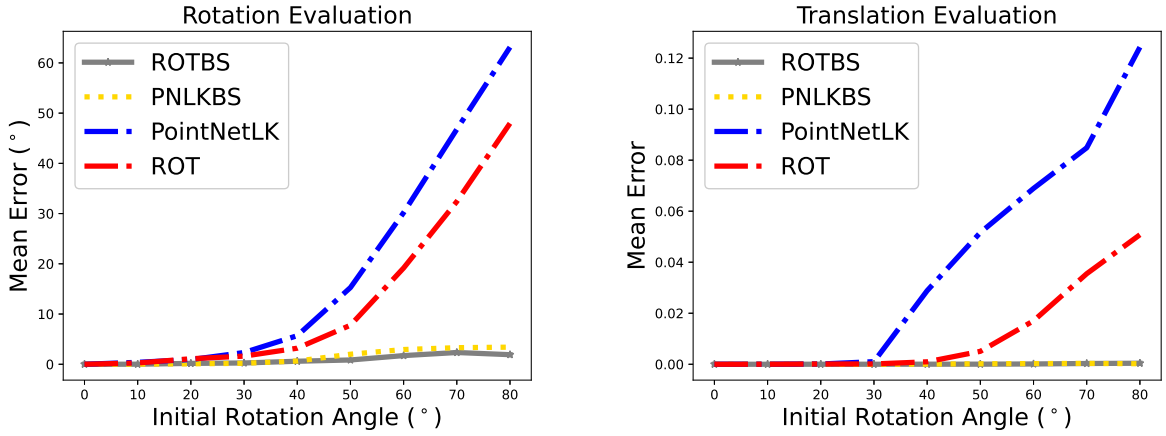


Figure 4.9: Example qualitative registration results for 3DMatch.



(a) (b)

Figure 4.10: Ablation Study: An evaluation of the impact of removing or modifying specific components or variables in a system. In this study, the vertical axis depicts the errors in rotation (a) and translation (b) that occur after the registration process. Meanwhile, the horizontal axis indicates the initial misalignment between the template and the source prior to registration.

4.3.8 Time complexity

This section also conducted an evaluation of the mean computational time of five popular point cloud alignment algorithms: Go-ICP, FGR, DCP, ROT, and ROTBS. The objective was to compare their performance in aligning two point clouds from the ModelNet40 dataset. The results of this evaluation are presented in Table 4.4. It can be seen that the average time required for aligning 1002 object pairs was calculated and compared among the algorithms.

It was observed that the proposed algorithms demonstrated faster performance compared to Go-ICP, with the number of points being approximately 2400 and the initial misalignment between the template and source point clouds being set at 90°. This indicates that the proposed algorithms have a significant advantage in terms of computational efficiency in aligning point clouds.

Table 4.4: Running time comparison with around 2.4K points.

Method	Go-ICP	FGR	ROT	ROTBS
Mean time (s)	89.01	0.15	0.11	12.28
Mean error (°)	4.79	13.70	59.63	3.19

4.4 Summary and conclusions

The chapter introduced an unsupervised end-to-end 3D rigid point cloud registration pipeline to tackle the issue of point cloud registration with large rotations and partial overlaps. This method adopts a rotation-based unsupervised approach to train networks that can effectively reduce the dependence on labeled data. To further tackle the challenge of large rotation registration, a beam search scheme was introduced, which can significantly enhance the performance of the method. Additionally, a clustering-based method was provided to solve the point cloud registration with partial overlaps. This search approach can be applied independently of the specific neural network architecture and can cooperate with other registration tasks that face large rotation challenges, even those that are correspondence-based.

However, despite its benefits, the method proposed in this chapter still faces challenges in real-world scenarios with low partial overlaps. For instance, on the 3DMatch dataset, the proposed method only achieved 89.14% registration performance. This is due to the difference in the feature extraction capabilities of PointNet. To address this issue, a correspondence-based method will be introduced in the following chapter to solve point cloud registration with low partial overlaps.

Chapter 5

FOTReg, Fused Optimal Transport based Point Cloud Registration

5.1 Introduction

Chapter 4 extends the correspondence-free registration methods to solve point cloud pairs with large rotations and partial overlaps, achieving outstanding performance on synthetic datasets. However, it underperforms on point clouds with low partial overlaps and real scenes, as discussed in Sec. 1.2.4. To register point clouds with partial overlaps, many works [9], [15], [150] have shown that the correspondence-based methods perform better than correspondence-free methods. However, the performance of correspondence-based approaches depends on the quality of the estimated correspondences. This chapter thus proposes a learning framework FOTReg by simultaneously considering pointwise and structural matchings to estimate more correct correspondences in the overlap regions.

Correspondence-based algorithms occupy a significant proportion of deep learning-based registration methods. The state-of-the-art correspondences-based pipelines commonly consist of the following stages [13]: *feature extraction, correspondence prediction, outlier rejection*, and *transformation estimation*. Correspondences-based approaches mainly focus on improving registration performance by extracting highly discriminative features [14], [15], [120], [127] or removing the outlier correspondences [12], [13], [152]. The correspondence prediction is also critical since estimating the transformation parameters depends on the correct correspondences. However, only a few works have been devoted to improving the correspondence prediction algorithms [12]. Therefore, this chapter focuses on developing a correspondence prediction algorithm to obtain more accurate correspondences for pairwise point cloud registration.

In learning-based 3D point registration, pointwise matching is often used to establish matches

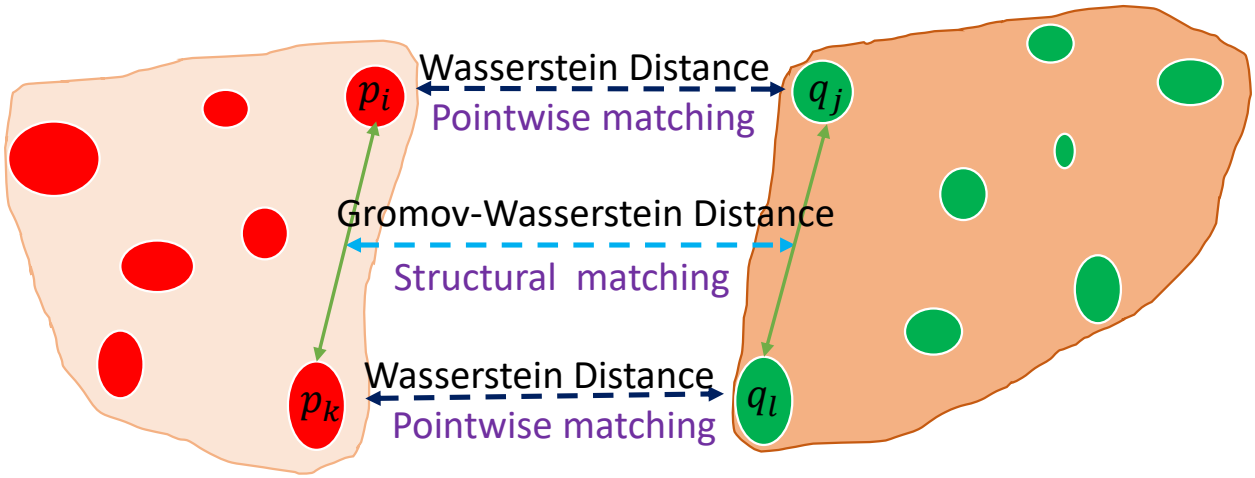


Figure 5.1: A paradigm shows the key concept of the proposed correspondence prediction algorithm. It consists of pointwise matching based on the feature similarities, such as $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$, and structural matching based on the geometric similarities between two pairs of points, such as $\{\boldsymbol{p}_i, \boldsymbol{p}_k\}$ and $\{\boldsymbol{q}_j, \boldsymbol{q}_l\}$. The size of each point represents its overlap score.

between two point clouds [31]. The core idea of pointwise matching is that a pair of points between the source and target point clouds having the most similar feature representations are identified as the corresponding points. However, the putative correspondences produced by the pointwise matching contain many false matches and outliers [211]. The following two factors will cause false matches: first of all, some points in the overlap regions are assigned to that in the non-overlap areas because they are treated equally in the correspondence prediction stage [14]; second, more than pointwise matching is required to distinguish the ambiguous and repeated patterns in the point clouds, hence estimating more correct correspondences [15].

To solve the above problems, borrowing ideas from Predator [14] and OCFNet [23], FOTReg first introduces the overlap scores to detect overlap regions, which then guide the correspondence prediction. Furthermore, inspired by the success of structural constraint, i.e., edge information, in graph matching [212], [213] and outlier rejection [13], structural matching as an additional condition is also applied to produce correspondences for registration. When combining overlap scores with pointwise and structural matchings, it induces a fused Gromov-Wasserstein distance (**FGW**) problem [163], which estimates the correspondences by searching a transport plan. The structure is defined as the distance between two points within a point cloud in Euclidean and feature spaces, so the structural matching is formulated by comparing the geometric difference based on the structural matching [136]. Figure 5.1 illustrates the pointwise and structural matchings. FOTReg is based on the intuition that incorporating pointwise and structural matchings into producing correspondences can resolve ambiguity, achieving better performance than using either match. Expressly, the FGW can incorporate both the pointwise and the structural matchings with the overlap scores acting as empirical distributions to improve the

accuracy of the generated correspondences in 3D point cloud registration. To ease the memory burden, the proposed method adopts the coarse-to-fine mechanism borrowed from CoFiNet [150] and a geotransformer [9], a hierarchical matching strategy to generate the correspondences.

5.2 Methodology

This section first explains the formulation and notation of the problem before introducing the proposed method. The problem involves two point sets, $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i \in \mathbb{R}^3 | i = 1, \cdots, N\}$ and $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_i \in \mathbb{R}^3 | i = 1, \cdots, M\}$, which have associated features $\boldsymbol{\mathcal{F}}_p = \{\boldsymbol{f}_{p_i} \in \mathbb{R}^d | i = 1, \cdots, N\}$ and $\boldsymbol{\mathcal{F}}_q = \{\boldsymbol{f}_{q_i} \in \mathbb{R}^d | i = 1, \cdots, M\}$, respectively. The objective of the registration problem is to find an optimal transformation $\boldsymbol{T}$ consisting of a rotation $\boldsymbol{R} \in SO(3)$ and a translation $\boldsymbol{t} \in \mathbb{R}^3$, which merges the source set $\boldsymbol{\mathcal{P}}$ with the target set $\boldsymbol{\mathcal{Q}}$. $\boldsymbol{T}$ can only be determined from the data in overlapping areas of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ if both sets have sufficient overlaps. An assignment matrix $\Gamma \in \mathbb{R}^{N \times M}$ with elements $\Gamma_{ij} \in \{0, 1\}$ is defined to represent the matching confidence between the point $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$, where each element satisfies

$$\Gamma_{ij} = \begin{cases} 1, & \text{if point } \boldsymbol{p}_i \text{ matches } \boldsymbol{q}_j \\ 0, & \text{otherwise} \end{cases}. \tag{5.1}$$

FOTReg uses the overlap scores to depict the overlapping region. The overlap score of the point $\boldsymbol{p}_i$ is defined as

$$\boldsymbol{\mu}_{\boldsymbol{p}_i} = \begin{cases} 1, & \text{if point } \boldsymbol{p}_i \text{ is an inlier of } \boldsymbol{\mathcal{P}} \\ 0, & \text{otherwise} \end{cases}, \tag{5.2}$$

where $\boldsymbol{\mu}_{\boldsymbol{q}_j}$ is defined similarly. Let $\boldsymbol{\mu}_p = \{\boldsymbol{\mu}_{\boldsymbol{p}_i} | i = 1, \cdots, N\}$ and $\boldsymbol{\mu}_q = \{\boldsymbol{\mu}_{\boldsymbol{q}_j} | j = 1, \cdots, M\}$. Usually, it is unlikely to determine whether a point is in overlap regions. Thus it needs to relax the overlap scores to $\boldsymbol{\mu}_{\boldsymbol{x}} \in [0, 1]$ and apply a neural network to predict overlap scores. The overlap score prediction module details are illustrated in Sec. 5.2.6. When considering the overlap scores, the registration task is then cast to solve the following problem:

$$\begin{aligned} &\min_{\Gamma, \boldsymbol{R}, \boldsymbol{t}} \sum_{i=1}^{N} \sum_{j=1}^{M} \Gamma_{ij} \| \boldsymbol{R}\boldsymbol{p}_i + \boldsymbol{t} - \boldsymbol{q}_j \|_2, \\ &\text{s.t.} \sum_{j=1}^{M} \Gamma_{ij} = \boldsymbol{\mu}_{\boldsymbol{p}_i}, \sum_{i=1}^{N} \Gamma_{ij} = \boldsymbol{\mu}_{\boldsymbol{q}_j}, \Gamma_{ij} \in \{0, 1\}. \end{aligned} \tag{5.3}$$

The three constraints enforce Γ to be a permutation matrix. If the optimal rigid transformation $\{\boldsymbol{R}, \boldsymbol{t}\}$ are provided, then the assignment matrix Γ can be recovered from Eq. (5.3). By contrary, given the assignment matrix Γ, correspondences can be estimated by $\mathcal{M} = \{(\boldsymbol{p}_i, \boldsymbol{q}_j) | j = \arg\max_k \Gamma_{ik}\}$. $\mathcal{M}$ can be directly leveraged by RANSAC [41] or SVD to estimate the transformation.

The following notation will be used throughout this chapter. $\boldsymbol{p}_i \rightarrow \boldsymbol{q}_j$ indicates a correspondence of $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$. $\{\boldsymbol{p}_i, \boldsymbol{p}_k\}$ represents a pair. $\langle \cdot, \cdot \rangle$ is an inner product operator. The Kullback-Leibler ($\mathcal{KL}$) divergence between two non-negative vectors $\boldsymbol{a} = (a_1, a_2, \cdots, a_n)$ and $\boldsymbol{b} = (b_1, b_2, \cdots, b_n)$ is defined as

$$\mathcal{KL}(\boldsymbol{a}|\boldsymbol{b}) = \sum_{i=1}^{n} \left(a_i \log \left(\frac{a_i}{b_i} \right) - a_i + b_i \right), \tag{5.4}$$

with the convention $0 \log 0 = 0$.

5.2.1 Optimal transport-based point cloud registration

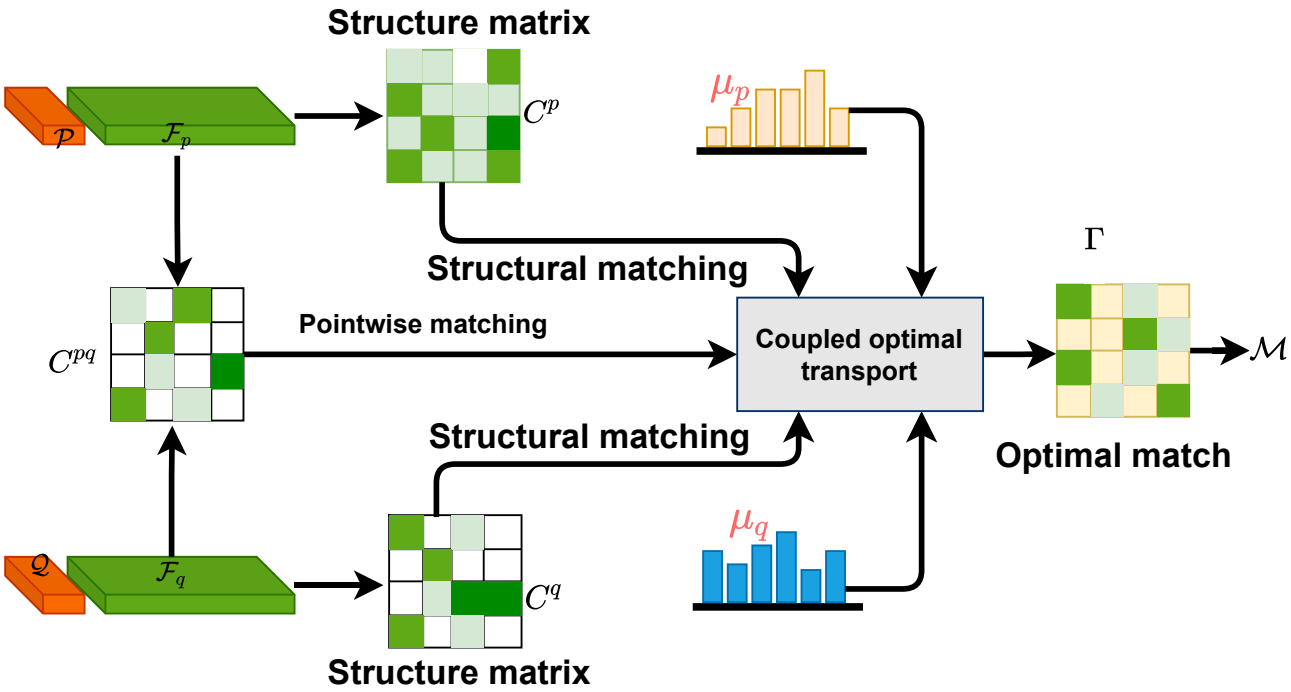


Figure 5.2: **Overview of the correspondence prediction.** Point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, with their features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$, and the overlap scores $\boldsymbol{\mu}_p, \boldsymbol{\mu}_q$. $\boldsymbol{C}^{pq}$ is the cross distance matrix. $\boldsymbol{C}^p$ and $\boldsymbol{C}^q$ represent the structure matrices. The assignment matrix Γ is predicted by solving a fused optimal transport problem. $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ have N and M points, respectively. $\mathcal{M}$ represents a set of estimated correspondences.

This section proposes a method (FOTReg) to generate correspondences by jointly considering pointwise and structural matchings, as illustrated in Figure 5.2. FOTReg first uses $\boldsymbol{\mathcal{F}}_p$ and the coordinates of $\boldsymbol{\mathcal{P}}$ to calculate $\boldsymbol{C}^p$. Similarly, $\boldsymbol{\mathcal{F}}_q$ and the coordinates of $\boldsymbol{\mathcal{Q}}$ are used to calculate the $\boldsymbol{C}^q$. After that, $\boldsymbol{C}^p$, $\boldsymbol{C}^q$, the overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$ are used for structural matching based on the Gromov-Wasserstein distance. $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ are used to calculate $\boldsymbol{C}^{pq}$. Then $\boldsymbol{C}^{pq}$ and overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$ are used for pointwise matching based on the Wasserstein distance. Finally, the Wasserstein distance and Gromov-Wasserstein distance are fused in a mutually-beneficial way by sharing the assignment matrix Γ, which is estimated via the fused optimal

transport algorithm. $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ have N and M points, respectively. The correspondences $\mathcal{M}$ are finally obtained based on $\mathcal{M} = \{(\boldsymbol{p}_i, \boldsymbol{q}_j)|j = \arg\max_k \Gamma_{ik}\}$. The following section will explain the utilization of fused optimal transport in predicting correspondences.

5.2.2 Fused optimal transport-based correspondence prediction

As mentioned above, only considering the pointwise matching is not sufficient to find accurate correspondences due to the ambiguous and repeated patterns in the 3D acquisition point clouds. Therefore, this section combines both pointwise and structural matchings to establish the correspondence between the source and target point clouds.

For point-wise matching, the cross-distance matrix $\boldsymbol{C}^{pq} \in \mathbb{R}^{N\times M}$ is derived based on the feature space between point cloud $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ with each element satisfying

$$\boldsymbol{C}_{ij}^{pq} = \mathcal{D}_f\left(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{q}_j}\right), 0 \leq i \leq N, 0 \leq j \leq M, \tag{5.5}$$

where $\mathcal{D}_f\left(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{q}_j}\right) = \|\frac{\boldsymbol{f}_{\boldsymbol{p}_i}}{\|\boldsymbol{f}_{\boldsymbol{p}_i}\|_2} - \frac{\boldsymbol{f}_{\boldsymbol{q}_j}}{\|\boldsymbol{f}_{\boldsymbol{q}_j}\|_2}\|_2$ represents the distance between $\boldsymbol{f}_{\boldsymbol{p}_i}$ and $\boldsymbol{f}_{\boldsymbol{q}_j}$.

To construct the structural matching, FOTReg first calculates the discrepancy (structure) matrix $\boldsymbol{C}^p \in \mathbb{R}^{N\times N}$ of the point pairs within $\boldsymbol{\mathcal{P}}$ in both Euclidean and feature spaces with the elements related to $\{\boldsymbol{p}_i, \boldsymbol{p}_k\}$ satisfying

$$\boldsymbol{C}_{ik}^{p} = \lambda \underbrace{\mathcal{D}_e\left(\boldsymbol{p}_i, \boldsymbol{p}_k\right)}_{Euclidean} + (1-\lambda)\underbrace{\mathcal{D}_f\left(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{p}_k}\right)}_{Feature}, \tag{5.6}$$

where $\mathcal{D}_e\left(\cdot,\cdot\right)$ is a function related to distances between two points in Euclidean space satisfying $\mathcal{D}_e\left(\boldsymbol{p}_i, \boldsymbol{p}_k\right) = 2\tanh(\|\boldsymbol{p}_i - \boldsymbol{p}_k\|_2)$. $\lambda \in [0,1]$ is a hyperparameter controlling the contribution of feature and coordinate information. Similarly, the elements of $\boldsymbol{C}^q \in \mathbb{R}^{M\times M}$ related to $\{\boldsymbol{q}_j, \boldsymbol{q}_l\}$ for $\boldsymbol{\mathcal{Q}}$ can be calculated by

$$\boldsymbol{C}_{jl}^{q} = \lambda\mathcal{D}_e\left(\boldsymbol{q}_j, \boldsymbol{q}_l\right) + (1-\lambda)\mathcal{D}_f\left(\boldsymbol{f}_{\boldsymbol{q}_j}, \boldsymbol{f}_{\boldsymbol{q}_l}\right). \tag{5.7}$$

The proposed method jointly exploit the pointwise and structural matchings equipped with overlap scores to indicate the correspondences between source and target point clouds. This leads to an optimization problem that can be solved by a fused optimal transport method (called fused optimal transport since it contains two types of distance, i.e., WD and GWD).

$$\begin{aligned} &\min_{\Gamma} \sum_{ij} \xi_1 \underbrace{\Gamma_{ij}\boldsymbol{C}_{ij}^{pq}}_{\text{WD}} + \xi_2 \sum_{ijkl} \underbrace{\Gamma_{ij}\Gamma_{k,l}\left(\boldsymbol{C}_{ik}^{p} - \boldsymbol{C}_{jl}^{q}\right)^2}_{\text{GWD}}, \\ &\text{s.t., } \Gamma\mathbf{1}_M = \boldsymbol{\mu}_p, \Gamma^\top\mathbf{1}_N = \boldsymbol{\mu}_q, \Gamma_{ij} \in [0,1], \end{aligned} \tag{5.8}$$

where $\mathbf{1}_n$ denotes an n-dimensional all-one vector. ξ_1 and ξ_2 are two non-negative hyperparameters controlling the pointwise and structural matchings, respectively. If $\xi_1 > 0$ and $\xi_2 = 0$,

then it only depends on the pointwise matching, and if $\xi_1 = 0$ and $\xi_2 > 0$, it only considers the structural matching. When $\xi_1 > 0$ and $\xi_2 > 0$, it allows the framework to effectively consider both pointwise and structural matchings for better correspondence predictions by sharing the assignment matrix Γ.

The next section will introduce the Wasserstein distance-based pointwise matching and the Gromov-Wasserstein distance-based structural matching, respectively.

5.2.3 Wasserstein distance-based pointwise matching

The Wasserstein distance-based pointwise matching method can be regarded as a variant of the nearest neighbor search with an additional bijectivity constraint that enforces global matching consistency. But it is appropriate for partial registration since the overlap scores have been introduced to detect the overlap regions. Given two point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, with their associated features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$, overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$, the assignment matrix Γ can be estimated based on the following Theorem.

Theorem 1. *Given features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$, overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$, if $\boldsymbol{\mathcal{F}}_p, \boldsymbol{\mathcal{F}}_q$ are invariant to rigid transformations, and $\boldsymbol{R}^\star, \boldsymbol{t}^\star, \Gamma^\star$ are the optimal solutions of problem in Eq.* (5.3)*. Then $\Gamma^\star$ is an optimal solution to the following optimization problem:*

$$\begin{aligned} &\min_{\Gamma} \langle \boldsymbol{C}^{pq}, \Gamma \rangle = \min_{\Gamma} \sum_{i=1}^{N} \sum_{j=1}^{M} \Gamma_{ij} \boldsymbol{C}_{ij}^{pq}, \\ &s.t.\ \Gamma \mathbf{1}_M = \boldsymbol{\mu}_p, \Gamma^\top \mathbf{1}_N = \boldsymbol{\mu}_q, \Gamma_{ij} \in [0,1], \end{aligned} \tag{5.9}$$

where $\boldsymbol{f}_{\boldsymbol{p}_i} \in \boldsymbol{\mathcal{F}}_p$, $\boldsymbol{f}_{\boldsymbol{q}_j} \in \boldsymbol{\mathcal{F}}_q$ represent the features of points $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$, respectively. $\boldsymbol{C}_{ij}^{pq} = \| \frac{\boldsymbol{f}_{\boldsymbol{p}_i}}{\|\boldsymbol{f}_{\boldsymbol{p}_i}\|_2} - \frac{\boldsymbol{f}_{\boldsymbol{q}_j}}{\|\boldsymbol{f}_{\boldsymbol{q}_j}\|_2} \|_2$. The constraint of the assignment matrix Γ is relaxed to a doubly stochastic state, that is, $\Gamma_{ij} \in [0,1]$.

Proof. This proof assumes the correct matches in the correspondence set are free of noise and denotes $h_1(\Gamma) = \langle \Gamma, \boldsymbol{C}^{pq} \rangle$. As the minimum values of $h_1(\Gamma)$ is non-negative, if $h_1(\Gamma^\star) = 0$ can be proved, then $\Gamma^\star$ is a optimal solution of (5.9). $\boldsymbol{R}^\star, \boldsymbol{t}^\star, \Gamma^\star$ are the optimal solutions of problem (5.3), leading to

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \Gamma_{ij}^\star \| \boldsymbol{R}^\star \boldsymbol{p}_i + \boldsymbol{t}^\star - \boldsymbol{q}_j \|_2^2 = 0$$

$$\Rightarrow \begin{cases} \| \boldsymbol{R}^\star \boldsymbol{p}_i + \boldsymbol{t}^\star - \boldsymbol{q}_j \|_2^2 = 0, & \Gamma_{ij}^\star = 1 \\ \Gamma_{ij}^\star \mathcal{D}_f \left(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{q}_j} \right) = 0, & \Gamma_{ij}^\star = 0 \end{cases},$$

i.e., when $\Gamma_{ij}^\star = 1$, $\boldsymbol{p}_i \to \boldsymbol{q}_j$ is an aligned correspondence since $\| \boldsymbol{R}^\star \boldsymbol{p}_i + \boldsymbol{t}^\star - \boldsymbol{q}_j \|_2^2 = 0$. As the features are invariant to rigid transformation, then

$$\boldsymbol{f}_{\boldsymbol{p}_i} = \boldsymbol{f}_{\boldsymbol{q}_j} \Rightarrow \Gamma_{ij}^\star \mathcal{D}_f \left(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{q}_j} \right) = 0 \Rightarrow h_1(\Gamma^\star) = 0. \tag{5.10}$$

□

Remark 1. *Theorem 1 signifies that the assignment matrix* Γ *can be calculated by solving the optimization problem in Eq.* (5.9)*, which is related to an optimal transport [136] problem (Wasserstein distance). It can be solved using the Sinkhorn algorithm [167].*

5.2.4 Gromov-Wasserstein distance-based structural matching

Structural matching is based on the following observations: for all $\boldsymbol{p}_i, \boldsymbol{p}_k \in \boldsymbol{\mathcal{P}}$ and $\boldsymbol{q}_j, \boldsymbol{q}_l \in \boldsymbol{\mathcal{Q}}$ with their associated features $\boldsymbol{fp}_i, \boldsymbol{fp}_k \in \boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{fq}_j, \boldsymbol{fq}_l \in \boldsymbol{\mathcal{F}}_q$, if the correspondences $\boldsymbol{p}_i \to \boldsymbol{q}_j$ and $\boldsymbol{p}_k \to \boldsymbol{q}_l$ are correct, then the distance between $\boldsymbol{p}_i$ and $\boldsymbol{p}_k$ should be similar to the distance between $\boldsymbol{q}_j$ and $\boldsymbol{q}_l$. This implies that the structural difference in both the Euclidean and feature spaces should be small, i.e., $|c^e(\boldsymbol{p}_i, \boldsymbol{p}_k) - c^e(\boldsymbol{q}_j, \boldsymbol{q}l)|$ and $|c^f(\boldsymbol{fp}i, \boldsymbol{fp}k) - c^f(\boldsymbol{fq}j, \boldsymbol{fq}_l)|$ should be small. The Gromov-Wasserstein distance is frequently utilized to determine the relationships between two sets of samples using their pairwise similarity (or distance) matrices within the domains. Thus, it can approximately transform the correspondence prediction into a structural matching problem based on the Gromov-Wasserstein distance, as stated in the following theorem.

Theorem 2. *Given two point clouds* $\boldsymbol{\mathcal{P}}$ *and* $\boldsymbol{\mathcal{Q}}$*, with their associated features* $\boldsymbol{\mathcal{F}}_p$ *and* $\boldsymbol{\mathcal{F}}_q$*, overlap scores* $\boldsymbol{\mu}_p$ *and* $\boldsymbol{\mu}_q$*, if* $\boldsymbol{\mathcal{F}}_p, \boldsymbol{\mathcal{F}}_q$ *are invariant to rigid transformation, and* $\boldsymbol{R}^\star, \boldsymbol{t}^\star, \Gamma^\star$ *are the optimal solutions of problem (5.3), then* $\Gamma^\star$ *is an optimal solution of the following Gromov-Wasserstein distance-based optimization:*

$$\begin{aligned} &\min_{\Gamma} \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{l=1}^{M} \Gamma_{ij}\Gamma_{kl}\left(\boldsymbol{C}_{ik}^{p} - \boldsymbol{C}_{jl}^{q}\right)^2 \\ &s.t.,\ \Gamma\mathbf{1}_M = \boldsymbol{\mu}_p, \Gamma^\top\mathbf{1}_N = \boldsymbol{\mu}_q, \Gamma_{ij} \in [0,1], \end{aligned} \tag{5.11}$$

where $\boldsymbol{C}_{ik}^{p}$ *and* $\boldsymbol{C}_{jl}^{q}$ *are defined as Eq.* (5.6) *and Eq.* (5.7)*, respectively.*

Proof. Denote $h_2(\Gamma) = \sum_{ijkl}\Gamma_{ij}\Gamma_{kl}\left(\boldsymbol{C}_{ik}^{p} - \boldsymbol{C}_{jl}^{q}\right)^2$. The minimum values of $h_2(\Gamma)$ are non-negative, $\Gamma^\star$ is an optimal solution of (5.11) can be translated into proving $h_2(\Gamma^\star) = 0$. If $\Gamma_{ij}^\star = 1$ and $\Gamma_{k,l} = 1$, then $\|\boldsymbol{R}^\star\boldsymbol{p}_i + \boldsymbol{t}^\star - \boldsymbol{q}_j\|_2^2 = 0$ and $\|\boldsymbol{R}^\star\boldsymbol{p}_k + \boldsymbol{t}^\star - \boldsymbol{q}_l\|_2^2 = 0 \Rightarrow \|\boldsymbol{p}_i - \boldsymbol{p}_k\|_2 = \|\boldsymbol{R}^\star\boldsymbol{p}_i + \boldsymbol{t}^\star - \boldsymbol{R}^\star\boldsymbol{p}_k + \boldsymbol{t}^\star\| = \|\boldsymbol{q}_j - \boldsymbol{q}_l\|_2$. Meantime, $\boldsymbol{p}_i$ matches $\boldsymbol{q}_k$ and $\boldsymbol{p}_j$ matches $\boldsymbol{q}_l$ imply $\boldsymbol{f}_{\boldsymbol{p}_i} = \boldsymbol{f}_{\boldsymbol{p}_k}$ and $\boldsymbol{f}_{\boldsymbol{q}_j} = \boldsymbol{f}_{\boldsymbol{q}_l}$, respectively, resulting in

$$\begin{aligned} \left(\boldsymbol{C}_{ik}^{p} - \boldsymbol{C}_{jl}^{q}\right)^2 =& [(1-\lambda)\mathcal{D}_f(\boldsymbol{f}_{\boldsymbol{p}_i}, \boldsymbol{f}_{\boldsymbol{p}_k}) + \lambda\mathcal{D}_e(\boldsymbol{p}_i, \boldsymbol{p}_k) \\ &-(1-\lambda)\mathcal{D}_f\left(\boldsymbol{f}_{\boldsymbol{q}_j}, \boldsymbol{f}_{\boldsymbol{q}_l}\right) - \lambda\mathcal{D}_e(\boldsymbol{q}_j, \boldsymbol{q}_l)]^2 \\ =& 0 \Rightarrow \Gamma_{ij}\Gamma_{kl}\left(\boldsymbol{C}_{ik}^{p} - \boldsymbol{C}_{jl}^{q}\right)^2 = 0. \end{aligned}$$

If $\Gamma_{ij}^\star = 0$ or $\Gamma_{k,l} = 0$, thus $h_2(\Gamma^\star) = 0$. □

Assignment matrix Γ can be obtained from by problem in Eq. (5.11) by solving an entropy regularized optimization. Structural matching is formulated by jointly considering the structural differences in both Euclidean and feature space.

5.2.5 Model optimization

This section introduces how to solve the problem in (5.8). For simplicity, a matrix $\boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma\right) \in \mathbb{R}^{N\times M}$ is denoted with each element satisfying:

$$\left[\boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma\right)\right]_{kl} = \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\boldsymbol{C}^p_{ik} - \boldsymbol{C}^q_{jl}\right)^2 \Gamma_{ij}. \tag{5.12}$$

Eq. (5.12) leads to

$$\sum_{ijkl}\Gamma_{ij}\Gamma_{kl}\left(\boldsymbol{C}^p_{ik} - \boldsymbol{C}^q_{jl}\right)^2 = \left\langle \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma\right), \Gamma\right\rangle. \tag{5.13}$$

The problem in Eq. (5.8) can be rewritten as

$$\min_{\Gamma\geq 0} \xi_1 \underbrace{\left\langle \boldsymbol{C}^{pq}, \Gamma\right\rangle}_{\text{WD}} + \xi_2 \underbrace{\left\langle \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma\right), \Gamma\right\rangle}_{\text{GWD}}, \tag{5.14a}$$

$$\text{s.t., } \Gamma \mathbf{1}_M = \boldsymbol{\mu}_p, \Gamma^\top \mathbf{1}_N = \boldsymbol{\mu}_q. \tag{5.14b}$$

Standard optimal transport only allows a meaningful comparison of measures with the same total mass, i.e., $\sum_{i=1}^{N}\boldsymbol{\mu}_{p_i} = \sum_{j=1}^{M}\boldsymbol{\mu}_{q_j}$, which does not always satisfy the registration requirement due to multiple correspondences. Following [214], the constraints in Eq. (5.14b) are replaced with soft-marginals ($\mathcal{KL}$ divergence). Optimization in Eq. (5.14) is then translated into an unconstrained approximate transport problem

$$\min_{\Gamma\geq 0} \xi_1 \left\langle \boldsymbol{C}^{pq}, \Gamma\right\rangle + \xi_2 \left\langle \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma\right), \Gamma\right\rangle + \tau\left(\mathcal{KL}\left(\Gamma\mathbf{1}_M | \boldsymbol{\mu}_p\right) + \mathcal{KL}\left(\Gamma^\top \mathbf{1}_N | \boldsymbol{\mu}_q\right)\right), \tag{5.15}$$

where $\tau > 0$ is a regularization parameter to adjust the strength of penalization of the soft margins. The generalized proximal point method [215] and projected gradient descent are adopted to solve the problem in Eq. (5.15) based on the $\mathcal{KL}$ metric. Following [161], [215], $\Gamma^{(k)}$ is fixed at iteration $k+1$ for $k \geq 0$, $\mathcal{KL}\left(\Gamma|\Gamma^{(k)}\right)$ acts as a regularization centered on the previous solution $\Gamma^{(k)}$. The update rule for Eq. (5.15) at iteration $k+1$ can be written as

$$\begin{aligned}\Gamma^{(k+1)} = \arg\min_{\Gamma\geq 0}\ & \epsilon\mathcal{KL}\left(\Gamma|\Gamma^{(k)}\right) + \xi_1\left\langle \boldsymbol{C}^{pq}, \Gamma\right\rangle + \xi_2\left\langle \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma^{(k)}\right), \Gamma\right\rangle \\ & + \tau\left(\mathcal{KL}\left(\Gamma\mathbf{1}_M | \boldsymbol{\mu}_p\right) + \mathcal{KL}\left(\Gamma^\top \mathbf{1}_N | \boldsymbol{\mu}_q\right)\right),\end{aligned} \tag{5.16}$$

with initialization $\Gamma^{(0)} = \boldsymbol{\mu}_p \boldsymbol{\mu}_q{}^\top$. $\epsilon > 0$ is a regularization parameter. $\mathcal{KL}\left(\Gamma|\Gamma^{(k)}\right)$ can be interpreted as a damping term that encourages $\Gamma^{(k+1)}$ not to be very far from $\Gamma^{(k)}$. For ϵ small enough, $\Gamma^{(k+1)}$ in Eq. (5.16) converges to the optimal solution of problem in Eq. (5.14) as τ increases. Choosing ϵ trades off convergence speed with closeness to the original transport problem [167]. The solution of the problem in Eq. (5.16) is based on the following theorem.

Theorem 3. *Denote* $f(\Gamma^{(k)}) = \xi_1 \boldsymbol{C}^{pq} + \xi_2 \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma^{(k)}\right) - \epsilon \log \Gamma^{(k)}$ *and* $\boldsymbol{C} \in \mathbb{R}^{N\times M}$ *with elements that satisfy* $\boldsymbol{C}_{ij} = [f\left(\Gamma^{(k)}\right)]_{ij}$. *The optimal solution for the objective in Eq.* (5.16) *can be obtained by solving the following dual entropic regularized objective,*

$$\begin{aligned}\min_{\boldsymbol{u},\boldsymbol{v}} h(\boldsymbol{u},\boldsymbol{v}) = \min_{\boldsymbol{u},\boldsymbol{v}} \epsilon \sum_{i=1}^{N}\sum_{j=1}^{M} \exp\left(\frac{\boldsymbol{u}_i + \boldsymbol{v}_j - \boldsymbol{C}_{ij}}{\epsilon}\right) \\ + \tau\left\langle \exp\left(-\frac{\boldsymbol{u}}{\tau}\right), \boldsymbol{\mu}_p\right\rangle + \tau\left\langle \exp\left(-\frac{\boldsymbol{v}}{\tau}\right), \boldsymbol{\mu}_q\right\rangle,\end{aligned} \tag{5.17}$$

where $\boldsymbol{u} \in \mathbb{R}^N, \boldsymbol{v} \in \mathbb{R}^M$ *are dual variables.*

Proof. This proof denotes

$$\begin{aligned} f(\Gamma^{(k)}) &= \xi_1 \boldsymbol{C}^{pq} + \xi_2 \boldsymbol{H}\left(\boldsymbol{C}^p, \boldsymbol{C}^q, \Gamma^{(k)}\right) - \epsilon \log \Gamma^{(k)}, \\ \mathcal{H}(\Gamma) &= \sum_{i=1}^{N}\sum_{j=1}^{M} \Gamma_{ij}(\log \Gamma_{ij} - 1). \end{aligned} \tag{5.18}$$

And then

$$\begin{aligned}\mathcal{KL}\left(\Gamma|\Gamma^{(k)}\right) &= \sum_{i=1}^{N}\sum_{j=1}^{M}\left(\Gamma_{ij}\log\left(\frac{\Gamma_{ij}}{\Gamma_{ij}^{(k)}}\right) - \Gamma_{ij} + \Gamma_{ij}^{(k)}\right) \\ &= \mathcal{H}(\Gamma) - \left\langle \log \Gamma^{(k)}, \Gamma\right\rangle + \mathbf{1}_N^\top \Gamma^{(k)} \mathbf{1}_M.\end{aligned}$$

After algebraic simplification, Eq. (5.16) can be rewritten as

$$\begin{aligned}\Gamma^{(k+1)} = \arg\min_{\Gamma\geq 0} &\left\langle f\left(\Gamma^{(k)}\right), \Gamma\right\rangle + \epsilon\mathcal{H}(\Gamma) \\ &+ \tau\left(\mathcal{KL}\left(\Gamma\mathbf{1}_M|\boldsymbol{\mu}_p\right) + \mathcal{KL}\left(\Gamma^\top\mathbf{1}_N|\boldsymbol{\mu}_q\right)\right) \\ &+ \epsilon\mathbf{1}_N^\top\Gamma^{(k)}\mathbf{1}_M,\end{aligned} \tag{5.19}$$

with initialization $\Gamma^{(0)} = \boldsymbol{\mu}_p\boldsymbol{\mu}_q{}^\top$. It can be solved iteratively with the help of the Sinkhorn-Knopp algorithm [167], [216]. For $\forall\epsilon > 0$, the problem (5.19) is strongly convex and lower semi-continuous. Meanwhile, $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$ are given non-negative vectors, strong duality and the existence of a minimizer for (5.19) is thus given by the Fenchel-Legendre dual form, which states that

$$\max_{\boldsymbol{u}\in\mathbb{R}^N, \boldsymbol{v}\in\mathbb{R}^M} -F^\star(-\boldsymbol{u}) - G^\star(-\boldsymbol{v}) - \epsilon\sum_{ij}\exp\left(\frac{u_i + v_j - \boldsymbol{C}_{ij}}{\epsilon}\right),$$

where $\boldsymbol{C}_{ij} = [f\left(\Gamma^{(n)}\right)]_{ij}$, and the function $F^\star(\cdot)$ and $G^\star(\cdot)$ take the following forms:

$$\begin{aligned} F^\star(\boldsymbol{u}) &= \sup_{\boldsymbol{z}\in\mathbb{R}^N} \boldsymbol{z}^\top\boldsymbol{u} - \tau\mathcal{KL}(\boldsymbol{z}|\boldsymbol{\mu}_p) \\ &= \tau\left\langle \exp\left(\frac{\boldsymbol{u}}{\tau}\right), \boldsymbol{\mu}_p\right\rangle - \boldsymbol{\mu}_p{}^\top\mathbf{1}_N, \\ G^\star(\boldsymbol{v}) &= \sup_{\boldsymbol{z}\in\mathbb{R}^M} \boldsymbol{z}^\top\boldsymbol{v} - \tau\mathcal{KL}(\boldsymbol{z}|\boldsymbol{\mu}_q) \\ &= \tau\left\langle \exp\left(\frac{\boldsymbol{v}}{\tau}\right), \boldsymbol{\mu}_q\right\rangle - \boldsymbol{\mu}_q{}^\top\mathbf{1}_M, \end{aligned} \tag{5.20}$$

Thus, it is proved by denoting $h(\boldsymbol{u},\boldsymbol{v}) = F^\star(-\boldsymbol{u}) + G^\star(-\boldsymbol{v}) + \epsilon\sum_{ij}\exp\left(\frac{\boldsymbol{u}_i+\boldsymbol{v}_j-\boldsymbol{C}_{ij}}{\epsilon}\right)$. □

The problem in Eq. (5.17) can be solved using the Sinkhorn algorithm [167], [214]. Specifically,

$$\frac{\partial h}{\partial \boldsymbol{u}} = 0 \Rightarrow \sum_{j}^{N} \exp\left(\frac{\boldsymbol{u}_i + \boldsymbol{v}_j - \boldsymbol{C}_{ij}}{\epsilon}\right) - \exp\left(-\frac{\boldsymbol{u}_i}{\tau}\right)\boldsymbol{\mu}_{\boldsymbol{p}_i} = 0$$
$$\Rightarrow \exp\left(\frac{\boldsymbol{u}_i}{\epsilon}\right)\sum_{j}^{N} \exp\left(\frac{\boldsymbol{v}_j - \boldsymbol{C}_{ij}}{\epsilon}\right) = \exp\left(-\frac{\boldsymbol{u}_i}{\tau}\right)\boldsymbol{\mu}_{\boldsymbol{p}_i}.$$

Let $(\boldsymbol{u}^k, \boldsymbol{v}^k, \boldsymbol{a}^k, \boldsymbol{b}^k)$ be the solution returned at the k-th iteration of the algorithm. Here $\boldsymbol{a} = B(\boldsymbol{u}, \boldsymbol{v})\mathbf{1}_M$ and $\boldsymbol{b} = B(\boldsymbol{u}, \boldsymbol{v})^\top \mathbf{1}_N$ with $B(\boldsymbol{u}, \boldsymbol{v}) = \mathrm{diag}\left(\exp\left(\frac{\boldsymbol{u}}{\epsilon}\right)\right) \cdot \exp\left(-\frac{\boldsymbol{C}}{\epsilon}\right) \cdot \mathrm{diag}\left(\exp\left(\frac{\boldsymbol{v}}{\epsilon}\right)\right)$. Suppose iteration $k+1$ for $k \geq 0$ has a fixed $\boldsymbol{v}^k$, i.e.,

$$\exp\left(\frac{\boldsymbol{u}_i^{k+1}}{\epsilon}\right)\sum_{j}^{N} \exp\left(\frac{\boldsymbol{v}_j^k - \boldsymbol{C}_{ij}}{\epsilon}\right) = \exp\left(-\frac{\boldsymbol{u}_i^{k+1}}{\tau}\right)\boldsymbol{\mu}_{\boldsymbol{p}_i}. \tag{5.21}$$

Multiplying both sides by $\exp\left(\frac{\boldsymbol{u}_i^k}{\epsilon}\right)$, Eq. (5.21) is translated into

$$\exp\left(\frac{\boldsymbol{u}_i^{k+1}}{\epsilon}\right)\boldsymbol{a}_i^k = \exp\left(\frac{\boldsymbol{u}_i^k}{\epsilon}\right)\exp\left(-\frac{\boldsymbol{u}_i^{k+1}}{\tau}\right)\boldsymbol{\mu}_{\boldsymbol{p}_i}$$
$$\Rightarrow \boldsymbol{u}^{k+1} = \left[\frac{\boldsymbol{u}^k}{\epsilon} + \log\left(\boldsymbol{\mu}_p\right) - \log\left(\boldsymbol{a}^k\right)\right]\frac{\epsilon\tau}{\epsilon+\tau}.$$

Similarly, with $\boldsymbol{u}^k$ fixed, Eq. (5.21) is translated into

$$\boldsymbol{v}^{k+1} = \left[\frac{\boldsymbol{v}^k}{\epsilon} + \log\left(\boldsymbol{\mu}_q\right) - \log\left(\boldsymbol{b}^k\right)\right]\frac{\epsilon\tau}{\epsilon+\tau}.$$

The pseudocode in Algorithm 4 illustrates the solution. The inner iterations can be determined by $\epsilon, \boldsymbol{C}_{ij}$ and $\max\{M, N\}$ and the proof is similar to Theorem 2 in [214]. In experiments, it can be found that when setting $\xi_1 = 1.0$, $\tau = 5.0$, $\epsilon = 0.001$, $N_I = 100$ and $N_O = 20$, it can obtain satisfactory results.

5.2.6 Combined with learning network

The solution of the optimization problem in Eq. (5.8) is sought over the space of $N \times M$ permutation matrices. Because of memory constraints and speed limitations, it is unsuitable for solving large-scale registration problems. To this end, FOTReg adopts a hierarchical matching strategy that establishes superpoint-level correspondences and then predicts point-level correspondences according to superpoint-level matches. The proposed FOTReg pipeline is illustrated in Fig. 5.3, which is a shared weighted two-stream encoder-decoder network. Given a pair of point cloud $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, the encoder aggregates the raw points into superpoints $\bar{\boldsymbol{\mathcal{P}}} = \{\bar{\boldsymbol{p}}_i \in \mathbb{R}^3 | i = 1, 2, ..., \bar{N}\}$ and $\bar{\boldsymbol{\mathcal{Q}}} = \{\bar{\boldsymbol{q}}_j \in \mathbb{R}^3 | j = 1, 2, ..., \bar{M}\}$, while jointly learning the associated features $\boldsymbol{\mathcal{F}}_{\bar{p}} = \{\boldsymbol{f}_{\bar{p}_i} \in \mathbb{R}^b | i = 1, 2, ..., \bar{N}\}$ and $\boldsymbol{\mathcal{F}}_{\bar{q}} = \{\boldsymbol{f}_{\bar{q}_j} \in \mathbb{R}^b | j = 1, 2, ..., \bar{M}\}$. The

Algorithm 4 Fused optimal transport algorithm.

Input: Distance matrices $\boldsymbol{C}^p$, $\boldsymbol{C}^q$ and $\boldsymbol{C}^{pq}$, overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$, and hyparameters $\tau, \epsilon > 0$, $\xi_1 = 1$, and the number of outer/inner iterations N_O, N_I.

Initialize $\Gamma^{(0)} = \boldsymbol{\mu}_p \boldsymbol{\mu}_q{}^\top$, $k = 0$.
for $k = 0 : N_O$ **do**
  $\xi_2 = \frac{k}{N_O}$
  Compute $\boldsymbol{C}_{ij} = \left[f\left(\Gamma^{(k)}\right) \right]_{ij}$
  while $k < N_I$ **do**
    $\boldsymbol{a}^k = B(\boldsymbol{u}^k, \boldsymbol{v}^k)\mathbf{1}_M, \boldsymbol{b}^k = B(\boldsymbol{u}^k, \boldsymbol{v}^k)^\top \mathbf{1}_N$.
    if k is even **then**
      $\boldsymbol{u}^{k+1} = \left[\frac{\boldsymbol{u}^k}{\epsilon} + \log\left(\boldsymbol{\mu}_p\right) - \log\left(\boldsymbol{a}^k\right) \right] \frac{\epsilon\tau}{\epsilon+\tau}$
      $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k$
    else
      $\boldsymbol{v}^{k+1} = \left[\frac{\boldsymbol{v}^k}{\epsilon} + \log\left(\boldsymbol{\mu}_q\right) - \log\left(\boldsymbol{b}^k\right) \right] \frac{\epsilon\tau}{\epsilon+\tau}$
      $\boldsymbol{u}^{k+1} = \boldsymbol{u}^k$
    end if
    $k = k + 1$.
  end while
  $\Gamma^{(k)} = B(\boldsymbol{u}^k, \boldsymbol{v}^k)$
  Output: $\Gamma^{(N_O)}$
end for

overlap attention block updates the features as $\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}}$, and projects them to coarse level overlap score vectors $\boldsymbol{\mu}_{\bar{p}} = \{\boldsymbol{\mu}_{\bar{p}_i} \in [0,1]\}_{i=1}^{\bar{N}}, \boldsymbol{\mu}_{\bar{q}} = \{\boldsymbol{\mu}_{\bar{q}_j} \in [0,1]\}_{i=1}^{\bar{M}}$. The updated features and overlap scores are then used to calculate the coarse-level correspondences. Finally, the decoder converts the superpoint level features and overlap scores into pointwise feature descriptors $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ and overlap scores $\boldsymbol{\mu}_p$ and $\boldsymbol{\mu}_q$. These are utilized to determine fine-level correspondences.

Encoder. Inspired by Predator [14], a shared KPConv [93], which consists of a series of ResNet-like blocks and stridden convolutions, simultaneously down-samples the raw point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ into superpoints $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{Q}}}$ and extracts associated features $\boldsymbol{\mathcal{F}}_{\bar{p}} = \{\boldsymbol{f}_{\bar{p}_j} \in \mathbb{R}^b | j = 1, 2, ..., \bar{N}\}$ and $\boldsymbol{\mathcal{F}}_{\bar{q}} = \{\boldsymbol{f}_{\bar{q}_j} \in \mathbb{R}^b | j = 1, 2, ..., \bar{M}\}$, respectively.

Overlap attention module. The overlap attention module estimates the probability (overlap score) of whether a point is in the overlapping area and consists of positional encoding, self-attention, cross-attention, and overlap score prediction. The positional encoding assigns intrinsic geometric properties to a pointwise feature, thus enhancing distinctions among point

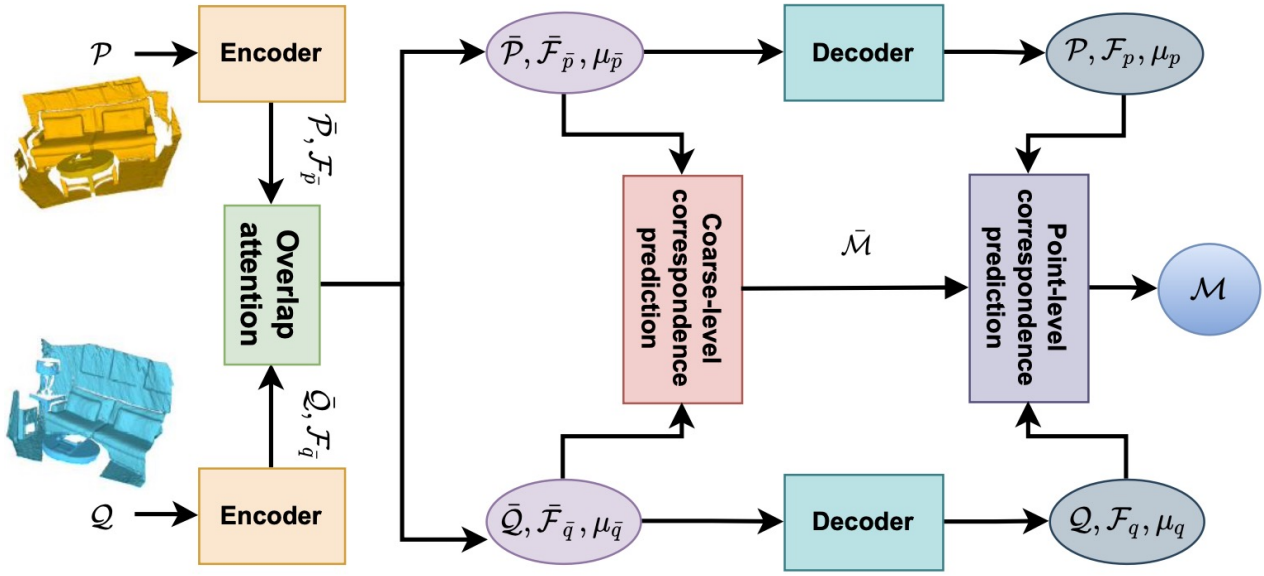


Figure 5.3: Overview of the proposed FOTReg combined with the network. FOTReg adopts a hierarchical matching strategy that establishes super point-level correspondences and then predicts point-level correspondences according to superpoint-level matches.

features in indistinctive regions. The extracted local features have a limited receptive field, which may not effectively distinguish indistinct regions. To address this issue, humans rely on the local neighborhood and a larger global context to identify correspondences in these indistinct regions. To model this long-range dependency, self-attention is introduced. The cross attention module further exploits the intra-relationship within the source and target point clouds, modeling the potential overlap regions. This section will provide further details on these individual components.

The positional encoding scheme assigns intrinsic geometric properties to per-point features by adding unique positional information, which improves the distinction of features in indistinctive regions. Given two superpoints $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{p}}_j$ from the set $\bar{\boldsymbol{\mathcal{P}}}$, the scheme selects the $k = 5$ nearest neighbors $\mathcal{K}_i$ of $\bar{\boldsymbol{p}}_i$ and calculates the centroid $\bar{\boldsymbol{p}}_c$ of $\bar{\boldsymbol{\mathcal{P}}}$, which is $\sum_{i=1}^{\bar{N}} \bar{\boldsymbol{p}}_i$. The angle between the vectors $\bar{\boldsymbol{p}}_i - \bar{\boldsymbol{p}}_c$ and $\bar{\boldsymbol{p}}_x - \bar{\boldsymbol{p}}_c$ is calculated for each $\bar{\boldsymbol{p}}_x \in \mathcal{K}_i$ and is denoted as α_{ix}. The position encoding $\boldsymbol{f}_{\bar{p}_i}^{pos}$ of $\bar{\boldsymbol{p}}_i$ is then defined as follows:

$$\boldsymbol{f}_{\bar{p}_i}^{pos} = \varphi\left(\|\bar{\boldsymbol{p}}_i - \bar{\boldsymbol{p}}_c\|_2\right) + \max_{x \in \mathcal{K}_i}\{\phi\left(\alpha_{ix}\right)\}, \tag{5.22}$$

where φ and ϕ are two MLPs, each MLP consists of a linear layer and one ReLU nonlinearity function.

Let $\boldsymbol{\mathcal{F}}_{\bar{p}}^{l}$ be the intermediate representation for $\bar{\boldsymbol{\mathcal{P}}}$ at layer l and let $\boldsymbol{\mathcal{F}}_{\bar{p}}^{0} = \{\boldsymbol{f}_{\bar{p}_i}^{pos} + \boldsymbol{f}_{\bar{p}_i}\}_{i=1}^{\bar{N}}$. A

Transformer module consisting of four parallel attention heads is applied to update the $\boldsymbol{\mathcal{F}}_{\bar{p}}^{l}$ via

$$\begin{aligned}
\boldsymbol{S}_{\bar{p}} &= \boldsymbol{W}_1^l \boldsymbol{\mathcal{F}}_{\bar{p}}^{l} + \boldsymbol{b}_1^l, \boldsymbol{K}_{\bar{x}} = \boldsymbol{W}_2^l \boldsymbol{\mathcal{F}}_{\bar{x}}^{l} + \boldsymbol{b}_2^l, \\
\boldsymbol{V}_{\bar{x}} &= \boldsymbol{W}_3^l \boldsymbol{\mathcal{F}}_{\bar{x}} + \boldsymbol{b}_3^l, \boldsymbol{A} = \operatorname{softmax}\left(\boldsymbol{S}_{\bar{p}}^{\top} \boldsymbol{K}_{\bar{x}} / \sqrt{b}\right), \\
\boldsymbol{\mathcal{F}}_{\bar{p}}^{l+1} &= \boldsymbol{\mathcal{F}}_{\bar{p}} + g^l\left(\left[\boldsymbol{\mathcal{F}}_{\bar{p}}^{l} \| \boldsymbol{A} \boldsymbol{V}_{\bar{x}}\right]\right).
\end{aligned} \tag{5.23}$$

Here, if $\bar{x} = \bar{p}$ represents self-attention, and if $\bar{x} = \bar{q}$ indicates cross-attention. $[\cdot\|\cdot]$ denotes concatenation, and $g^l(\cdot)$ is a three-layer fully connected network consisting of a linear layer, instance normalization, and a LeakyReLU activation. The same attention module is also simultaneously performed for all points in point cloud $\bar{\boldsymbol{\mathcal{Q}}}$. A fixed number of layers $L = 2$ with different parameters are chained and alternatively aggregate along the self- and cross-attention. As such, start from $l = 0, \bar{x} = \bar{p}$ if l is even and $\bar{x} = \bar{q}$ if l is odd. The final outputs of attention module are $\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}} = \boldsymbol{\mathcal{F}}_{p}^{3}$ for $\bar{\boldsymbol{\mathcal{P}}}$ and $\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}} = \boldsymbol{\mathcal{F}}_{\bar{q}}^{3}$ for $\bar{\boldsymbol{\mathcal{Q}}}$. By doing this, each point can incorporate non-local information that intuitively strengthens their long-range correlation dependencies. The latent features $\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ have the knowledge of $\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}}$ and vice versa.

Overlap score prediction. To deal with those points in non-overlapping regions, this module separately predicts the overlap scores $\boldsymbol{\mu}_{\bar{p}} = \{\boldsymbol{\mu}_{\bar{p}_1}, \boldsymbol{\mu}_{\bar{p}_2}, \cdots, \boldsymbol{\mu}_{\bar{p}_{\bar{N}}}\}$ and $\boldsymbol{\mu}_{\bar{q}} = \{\boldsymbol{\mu}_{\bar{q}_1}, \boldsymbol{\mu}_{\bar{q}_2}, \cdots, \boldsymbol{\mu}_{\bar{q}_{\bar{M}}}\}$ using the conditioned features $\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}}$ and $\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}}$. Here $\boldsymbol{\mu}_{\bar{p}_i} \in [0, 1]$ and $\boldsymbol{\mu}_{\bar{q}_j} \in [0, 1]$ can be computed using a single fully layer $g_\beta(\cdot)$ followed by a sigmoid activation function.

$$\begin{aligned}
\boldsymbol{\mu}_{\bar{p}} &= \operatorname{sigmoid}\left(g_\beta\left(\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}}\right)\right), \\
\boldsymbol{\mu}_{\bar{q}} &= \operatorname{sigmoid}\left(g_\beta\left(\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}}\right)\right).
\end{aligned} \tag{5.24}$$

The overlap scores mask the influence of points outside the overlap region.

Coarse-Level Correspondence Prediction. $\bar{\boldsymbol{\mathcal{F}}}_{\bar{p}}$, $\bar{\boldsymbol{\mathcal{F}}}_{\bar{q}}$, $\boldsymbol{\mu}_{\bar{p}}$ and $\boldsymbol{\mu}_{\bar{q}}$ are used to calculate an assignment matrix $\bar{\boldsymbol{\Gamma}}$, at superpoint level, by solving a fused optimal transport problem

$$\begin{aligned}
\min_{\bar{\boldsymbol{\Gamma}} \geq 0} & \left\langle \xi_1 \boldsymbol{C}^{\bar{p}\bar{q}}, \bar{\boldsymbol{\Gamma}} \right\rangle + \left\langle \xi_2 \boldsymbol{H}\left(\boldsymbol{C}^{\bar{p}}, \boldsymbol{C}^{\bar{q}}, \bar{\boldsymbol{\Gamma}}\right), \bar{\boldsymbol{\Gamma}} \right\rangle \\
& + \tau\left(\mathcal{KL}\left(\bar{\boldsymbol{\Gamma}} \mathbf{1}_{\bar{M}} | \boldsymbol{\mu}_{\bar{p}}\right) + \mathcal{KL}\left(\bar{\boldsymbol{\Gamma}}^{\top} \mathbf{1}_{\bar{N}} | \boldsymbol{\mu}_{\bar{q}}\right)\right),
\end{aligned} \tag{5.25}$$

where $\boldsymbol{C}^{\bar{p}\bar{q}}, \boldsymbol{C}^{\bar{p}}$, and $\boldsymbol{C}^{\bar{q}}$ are the distance matrices with elements satisfying $\boldsymbol{C}_{ij}^{\bar{p}\bar{q}} = \mathcal{D}_f\left(\bar{\boldsymbol{f}}_{\bar{p}_i}, \bar{\boldsymbol{f}}_{\bar{q}_j}\right)$, $\boldsymbol{C}_{ij}^{\bar{p}} = \lambda D_e\left(\bar{p}_i, \bar{p}_j\right) + (1-\lambda)\mathcal{D}_f\left(\bar{\boldsymbol{f}}_{\bar{p}_i}, \bar{\boldsymbol{f}}_{\bar{p}_j}\right)$, and $\boldsymbol{C}_{ij}^{\bar{q}} = \lambda D_e\left(\bar{q}_i, \bar{q}_j\right) + (1-\lambda)\mathcal{D}_f\left(\bar{\boldsymbol{f}}_{\bar{q}_i}, \bar{\boldsymbol{f}}_{\bar{q}_j}\right)$, respectively. $\lambda > 0$ is a hyperparameter. Eq. (5.25) is an instance of the optimal transport [167] problem, which can be solved efficiently using the Sinkhorn-Knopp algorithm [167]. Upon reaching $\bar{\boldsymbol{\Gamma}}$, the correspondences with the highest confidence score in each row and column can be selected and further refined through the application of the mutual nearest neighbor (MNN) criterion to eliminate potential outliers in the coarse matches. These correspondences coarse-level correspondences are defined as follows:

$$\bar{\mathcal{M}} = \left\{\left(\bar{\boldsymbol{p}}_{\hat{i}}, \bar{\boldsymbol{p}}_{\hat{j}}\right) \middle| \forall(\hat{i}, \hat{j}) \in \operatorname{MNN}(\bar{\boldsymbol{\Gamma}}), \hat{j} = \arg\max_k \bar{\boldsymbol{\Gamma}}_{\hat{i}k}\right\}. \tag{5.26}$$

Decoder. The decoder starts with conditioned features $\boldsymbol{\mathcal{F}}_{\bar{p}}$, concatenates them with the overlap score $\boldsymbol{\mu}_{\bar{p}}$, and outputs the pointwise feature descriptor $\boldsymbol{\mathcal{F}}_p \in \mathbb{R}^{N\times 32}$ and refined per-point overlap scores $\boldsymbol{\mu}_p$. The architecture of the decoder integrates NN-upsampling with linear layers and incorporates skip connections from the corresponding layers in the encoder. The same operator is applied to generate $\boldsymbol{\mathcal{F}}_q \in \mathbb{R}^{M\times 32}$ and $\boldsymbol{\mu}_q$.

Fine-level prediction. The finer stage refines coarse correspondences to point-level correspondences. Those refined matches are then utilized for point cloud registration. The points are first grouped into clusters by assigning points to their nearest superpoints in geometry space. After grouping, points with their corresponding overlap scores and descriptors form patches, on which point correspondences can be extracted. For each superpoint $\bar{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}$, its associated point set $G_{\bar{p}_i}$, feature set $G_{\boldsymbol{f}_{\bar{p}_i}}$, and the overlap score set $G_{\boldsymbol{\mu}_{\bar{p}_i}}$ are denoted as

$$\begin{cases} G_{\bar{p}_i} = \{\boldsymbol{p} \in \boldsymbol{\mathcal{P}} \big| \|\boldsymbol{p} - \bar{\boldsymbol{p}}_i\|_2 \leq \|\boldsymbol{p} - \bar{\boldsymbol{p}}_j\|_2, i \neq j\}, \\ G_{\boldsymbol{f}_{\bar{p}_i}} = \{\boldsymbol{f}_{\boldsymbol{x}_j} \in \boldsymbol{\mathcal{F}}_p \big| \boldsymbol{x}_j \in G_{\bar{p}_i}\}, \\ G_{\boldsymbol{\mu}_{\bar{p}_i}} = \{\mu_{\boldsymbol{x}_j} \in \boldsymbol{\mu}_p \big| \boldsymbol{x}_j \in G_{\bar{p}_i}\}. \end{cases} \tag{5.27}$$

The coarse-level correspondence set $\bar{\mathcal{M}}$ is expanded to its associated patch correspondence sets, both in geometry space $\mathcal{M}_C = \{(G_{\bar{p}_i}, G_{\bar{q}_j})\}$, feature space $\mathcal{M}_F = \{(G_{\boldsymbol{f}_{\bar{p}_i}}, G_{\boldsymbol{f}_{\bar{q}_j}})\}$, and overlap scores $\mathcal{M}_{\boldsymbol{\mu}} = \{(G_{\boldsymbol{\mu}_{\bar{p}_i}}, G_{\boldsymbol{\mu}_{\bar{q}_j}})\}$. For computational efficiency, every patch samples K points based on the overlap scores. Given a pair of overlapped patches $(G_{\bar{p}_i}, G_{\boldsymbol{f}_{\bar{p}_i}}, G_{\boldsymbol{\mu}_{\bar{p}_i}})$ and $(G_{\bar{q}_j}, G_{\boldsymbol{f}_{\bar{q}_j}}, G_{\boldsymbol{\mu}_{\bar{q}_j}})$, it needs to first calculate the cross distance matrix $\boldsymbol{C}^{\bar{p}_i\bar{q}_j} = \{\boldsymbol{C}_{kl}^{\bar{p}_i\bar{q}_j}\}$, and structural matrices $\boldsymbol{C}^{\bar{p}_i} = \{\boldsymbol{C}_{kl}^{\bar{p}_i}\}$ and $\boldsymbol{C}^{\bar{q}_j} = \{\boldsymbol{C}_{kl}^{\bar{q}_j}\}$ with elements satisfying

$$\begin{aligned} \boldsymbol{C}_{kl}^{\bar{p}_i\bar{q}_j} &= \mathcal{D}_f\left(G^k_{\boldsymbol{f}_{\bar{p}_i}}, G^l_{\boldsymbol{f}_{\bar{q}_j}}\right), \\ \boldsymbol{C}_{kl}^{\bar{p}_i} &= \lambda\mathcal{D}_e\left(G^k_{\bar{p}_i}, G^l_{\bar{p}_i}\right) + (1-\lambda)\mathcal{D}_f\left(G^k_{\boldsymbol{f}_{\bar{p}_i}}, G^l_{\boldsymbol{f}_{\bar{p}_i}}\right), \\ \boldsymbol{C}_{kl}^{\bar{q}_j} &= \lambda\mathcal{D}_e\left(G^k_{\bar{q}_j}, G^l_{\bar{q}_j}\right) + (1-\lambda)\mathcal{D}_f\left(G^k_{\boldsymbol{f}_{\bar{q}_j}}, G^l_{\boldsymbol{f}_{\bar{q}_j}}\right), \end{aligned}$$

where $\lambda \in [0,1]$ is a hyperparameter. Extracting point correspondences is analogous to matching two smaller-scale point clouds by solving a fused optimal transport problem to calculate a matrix $\Gamma_{\bar{p}_i}$ as

$$\begin{aligned} \min_{\Gamma_{\bar{p}_i}\geq 0} \langle \xi_1 \boldsymbol{C}^{\bar{p}_i\bar{q}_j}, \boldsymbol{\Gamma}_{\bar{p}_i}\rangle + \langle \xi_2 \boldsymbol{H}\left(\boldsymbol{C}^{\bar{p}_i}, \boldsymbol{C}^{\bar{q}_j}, \boldsymbol{\Gamma}_{\bar{p}_i}\right), \boldsymbol{\Gamma}_{\bar{p}_i}\rangle \\ + \tau\left(\mathcal{KL}\left(\boldsymbol{\Gamma}_{\bar{p}_i}\mathbf{1}_{M_{\bar{q}_j}} | G_{\boldsymbol{\mu}_{\bar{p}_i}}\right) + \mathcal{KL}\left(\boldsymbol{\Gamma}_{\bar{p}_i}^\top \mathbf{1}_{N_{\bar{q}_j}} | G_{\boldsymbol{\mu}_{\bar{q}_j}}\right)\right), \end{aligned}$$

For correspondences, the maximum confidence score of $\Gamma_{\bar{p}_i}$ is selected in each row and column to ensure higher precision. The final set of point correspondences $\mathcal{M}$ is the combination of all the obtained correspondence sets. After obtaining the correspondences $\mathcal{M}$, following [9], [150], a variant of RANSAC [41] that is specialized to 3D correspondence-based registration [217] is utilized to estimate the transformation.

5.2.7 Loss function and training

The proposed model is an end-to-end learning framework that utilizes ground truth correspondences as supervision. The loss function $\mathcal{L} = \mathcal{L}_C + \mathcal{L}_F + \mathcal{L}_{CO} + \mathcal{L}_{FO}$ consists of a coarse-level loss $\mathcal{L}_C$ for superpoint matching, a point matching loss $\mathcal{L}_F$ for point matching, a binary classification loss $\mathcal{L}_{CO}$ for coarse-level overlap scores, and a classification loss $\mathcal{L}_{FO}$ for fine-level overlap scores.

Superpoint matching loss. Existing methods [15], [150] usually formulate superpoint matching as a multilabel classification problem and adopt a cross-entropy loss with optimal transport. Doing this requires unfolding the Sinkhorn layer to compute gradients in the training stage. To address this issue, a circle loss [218] is adopted to optimize the superpoint-wise feature descriptors. As there is not direct supervision for superpoint matching, the overlap ratio r_i^j of points in $G_{\bar{p}_i}$ that have correspondences in $G_{\bar{q}_j}$ are used to depict the matching probability between superpoints $\bar{p}_i$ and $\bar{q}_j$. r_i^j is defined as:

$$r_i^j = \frac{1}{|G_{\bar{p}_i}|} |\{\boldsymbol{p} \in G_{\bar{p}_i} \big| \min_{\boldsymbol{q} \in G_{\bar{q}_j}} \|\hat{\boldsymbol{T}}(\boldsymbol{p}) - \boldsymbol{q}\|_2 < r_p\}|.$$

where $\hat{\boldsymbol{T}}$ is the ground-truth transformation and r_p is a set threshold. For circle loss, a positive pair of superpoints is defined as those whose corresponding patches have at least 10% overlap, while a negative pair is defined as those that do not overlap. Pairs that do not fall into either category are omitted. From the set of superpoints in $\bar{\boldsymbol{\mathcal{P}}}$, those with at least one positive superpoint in $\bar{\boldsymbol{\mathcal{Q}}}$ are selected to form the set of anchor superpoints, $\tilde{\boldsymbol{\mathcal{P}}}$. For each anchor $\tilde{\boldsymbol{p}}_i$ in $\tilde{\boldsymbol{\mathcal{P}}}$, the set of its positive superpoints in $\bar{\boldsymbol{\mathcal{Q}}}$ is designated as $\mathcal{N}_p^{\tilde{\boldsymbol{p}}_i}$, and the set of its negative patches is designated as $\mathcal{N}_n^{\tilde{\boldsymbol{p}}_i}$. The superpoint matching loss(circle loss) $\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}}$ on $\bar{\boldsymbol{\mathcal{P}}}$ is calculated by

$$\begin{aligned} \mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}} &= \frac{1}{|\tilde{\boldsymbol{\mathcal{P}}}|} \sum_{\tilde{\boldsymbol{p}}_i \in \tilde{\boldsymbol{\mathcal{P}}}} \log\left[1 + \zeta_i\right], \\ \zeta_i &= \sum_{\tilde{\boldsymbol{q}}_k \in \mathcal{N}_p^{\tilde{\boldsymbol{p}}_i}} e^{r_i^k \beta_p^{ik} (d_i^k - \Delta p)} \cdot \sum_{\tilde{\boldsymbol{q}}_l \in \mathcal{N}_n^{\tilde{\boldsymbol{p}}_i}} e^{\beta_n^{il} (\Delta n - d_i^l)}, \end{aligned} \tag{5.28}$$

where $d_i^k = \mathcal{D}_f(\boldsymbol{f}_{\tilde{p}_i}, \boldsymbol{f}_{\tilde{q}_k})$ denotes the distance in the feature space. The empirical margins are used to calculate the weights β_p^{ik} and β_n^{il} individually for each positive and negative example associated with $\Delta p = 0.1$ and $\Delta n = 1.4$, with a learned scale factor, γ, which must be greater than or equal to 1. The circle loss adjusts the loss values on $\mathcal{N}_p^{\tilde{\boldsymbol{p}}_i}$ according to the overlap ratio, giving higher importance to patch pairs with a higher overlap. The same is true for the loss $\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{Q}}}}$ on $\bar{\boldsymbol{\mathcal{Q}}}$. The overall superpoint matching loss is

$$\mathcal{L}_C = \frac{1}{2}(\mathcal{L}_C^{\bar{\boldsymbol{\mathcal{P}}}} + \mathcal{L}_C^{\bar{\boldsymbol{\mathcal{Q}}}}). \tag{5.29}$$

Coarse-level overlap loss. The ratio of points in $G_{\bar{p}_i}$ that are visible in $\mathcal{Q}$ are utilized to depict the ground-truth overlap scores $\bar{\boldsymbol{\mu}}_{\bar{p}_i}$ of $\bar{p}_i$. It is calculated by

$$\bar{\boldsymbol{\mu}}_{\bar{p}_i} = \frac{1}{|G_{\bar{p}_i}|}|\{\boldsymbol{p} \in G_{\bar{p}_i} | \min_{\boldsymbol{q}\in\mathcal{Q}} \|\hat{\boldsymbol{T}}(\boldsymbol{p}) - \boldsymbol{q}\|_2 < r_o\}|, \tag{5.30}$$

with overlap threshold. If $\bar{\boldsymbol{\mu}}_{\bar{p}_i}$ is close to 1, $\bar{p}_i$ tends to locate in the overlap regions. $\bar{\boldsymbol{\mu}}_{\bar{q}_j}$ is calculated in the same way. The predicted overlap scores for $\bar{\boldsymbol{\mathcal{P}}}$ are thus supervised using the binary cross-entropy loss, i.e.,

$$\mathcal{L}_{\bar{\boldsymbol{\mathcal{P}}}} = -\frac{1}{\bar{N}} \sum_i \bar{\boldsymbol{\mu}}_{\bar{p}_i} \log \boldsymbol{\mu}_{\bar{p}_i} + (1 - \bar{\boldsymbol{\mu}}_{\bar{p}_i}) \log (1 - \boldsymbol{\mu}_{\bar{p}_i}) . \tag{5.31}$$

The loss $\mathcal{L}_{\bar{\boldsymbol{\mathcal{Q}}}}$ for $\bar{\boldsymbol{\mathcal{Q}}}$ is calculated in the same way. The loss for coarse-level overlap scores is

$$\mathcal{L}_{CO} = \frac{1}{2} (\mathcal{L}_{\bar{\boldsymbol{\mathcal{P}}}} + \mathcal{L}_{\bar{\boldsymbol{\mathcal{Q}}}}) .$$

Point matching loss. Circle loss is applied again to supervise the point matching. Consider a pair of matched superpoints $\bar{\boldsymbol{p}}_i$ and $\bar{\boldsymbol{q}}_j$ with associated patches $G_{\bar{p}_i}$ and $G_{\bar{q}_j}$, it needs to first extract a set of anchor points $\tilde{G}_{\bar{p}_i} \subseteq G_{\bar{p}_i}$ satisfying that each $\boldsymbol{g}^k_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}$ has at least one (possibly multiple) correspondence in $G_{\bar{q}_j}$, i.e.,

$$\tilde{G}_{\bar{p}_i} = \{\boldsymbol{g}^k_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i} | \min_{\boldsymbol{g}^l_{\bar{q}_j} \in G_{\bar{q}_j}} \|\hat{\boldsymbol{T}}\left(\boldsymbol{g}^k_{\bar{p}_i}\right) - \boldsymbol{g}^l_{\bar{q}_j}\|_2 < r_p\}.$$

For each anchor $\boldsymbol{g}^k_{\bar{p}_i}$ in $\tilde{G}_{\bar{p}_i}$, the set of positive points in $G\bar{q}_j$ is denoted as $\mathcal{N}_p^{\boldsymbol{g}^k_{\bar{p}_i}}$. The set of negative patches is formed by all points in $\mathcal{Q}$ outside a larger radius r_n. The fine-level matching loss $\mathcal{L}_F^{\mathcal{P}}$ on $\mathcal{P}$ is calculated as:

$$\begin{aligned} \mathcal{L}_F^{\mathcal{P}} &= \frac{1}{|\tilde{\boldsymbol{\mathcal{P}}}|} \sum_{\tilde{\boldsymbol{p}}_i \in \bar{\boldsymbol{\mathcal{P}}}} \frac{1}{|\tilde{G}_{\bar{p}_i}|} \sum_{\boldsymbol{g}^s_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}} \log \left[1 + \xi_s\right], \\ \xi_s &= \sum_{\boldsymbol{g}^k_{\bar{q}_j} \in \mathcal{N}_p^{\boldsymbol{g}^s_{\bar{p}_i}}} e^{r^k_s \beta^{sk}_p (d^k_s - \Delta p)} \cdot \sum_{\boldsymbol{g}^l_{\bar{q}_j} \in \mathcal{N}_n^{\boldsymbol{g}^s_{\bar{p}_i}}} e^{\beta^{sl}_n (\Delta n - d^l_s)}, \end{aligned} \tag{5.32}$$

where $d^k_s = \mathcal{D}_f(\boldsymbol{f}_{\boldsymbol{g}^s_{\bar{p}_i}}, \boldsymbol{f}_{\boldsymbol{g}^s_{\bar{q}_j}})$ is the distance of two features. The weights $\beta^{sk}_p = \omega d^k_s$ and $\beta^{sl}_n = \omega(2.0 - d^l_s)$ for each positive and negative example are calculated separately with a learned scaling factor $\omega \geq 1$. The value of Δp is set to 0.1 and Δn is set to 1.4. The same is applied to the loss $\mathcal{L}_F^{\mathcal{Q}}$ for $\boldsymbol{\mathcal{Q}}$. The overall superpoint matching loss writes as

$$\mathcal{L}_F = \frac{1}{2}(\mathcal{L}_F^{\mathcal{P}} + \mathcal{L}_F^{\mathcal{Q}}). \tag{5.33}$$

Fine-level overlap loss. The overlap score loss $\mathcal{L}_{FO}$ is

$$\begin{aligned}\mathcal{L}_{FO} &= -\frac{1}{2}\left(\frac{1}{|\bar{\boldsymbol{\mathcal{P}}}|}\sum_{\bar{p}_i}\mathcal{L}_{\bar{p}_i} + \frac{1}{|\bar{\boldsymbol{\mathcal{Q}}}|}\sum_{\bar{q}_j}\mathcal{L}_{\bar{q}_j}\right), \\ \mathcal{L}_{\bar{p}_i} &= \frac{1}{|\tilde{G}_{\bar{p}_i}|}\sum_{\boldsymbol{g}^k_{\bar{p}_i}}\left(\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}\log\boldsymbol{\mu}_{\boldsymbol{g}^k_{\bar{p}_i}} + \left(1-\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}\right)\log\left(1-\boldsymbol{\mu}_{\boldsymbol{g}^k_{\bar{p}_i}}\right)\right).\end{aligned} \tag{5.34}$$

The ground-truth label $\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}}$ of the point $\boldsymbol{g}^k_{\bar{p}_i} \in \tilde{G}_{\bar{p}_i}$ is defined as

$$\bar{\boldsymbol{\mu}}_{\boldsymbol{g}^k_{\bar{p}_i}} = \begin{cases} 1, & \left(\min_{q_j\in\boldsymbol{\mathcal{Q}}}\|\hat{\boldsymbol{T}}(\boldsymbol{g}^k_{\bar{p}_i}) - \boldsymbol{q}_j\|\right) < r_o \\ 0, & \text{otherwise} \end{cases}, \tag{5.35}$$

where $\mathcal{L}_{\bar{q}_j}$ is calculated in the same way.

5.3 Experiments

This section conducts extensive experiments to evaluate the performance of the proposed method on indoor 3DMatch [31] and 3DLoMatch [14] benchmarks, outdoor KITTI [219] benchmark, and cross-source 3DCSR [22] benchmark.

5.3.1 Implementation details

The proposed method is implemented in PyTorch and can be trained on a single Quadro GV100 GPU (32G) and two Intel(R) Xeon(R) Gold 6226 CPUs. The hyperparameters are set as follows: $\xi_1 = 1.0$, $\tau = 5.0, \lambda = 0.1$, $\epsilon = 0.001$, $N_I = 100$, and $N_O = 20$. In all experiments, FOTReg is trained for 120 epochs with a batch size of 1 using the ADAM optimizer and an initial learning rate of $5e-4$ with a decaying factor of 0.99. The encoder and decoder architectures are the same as those used in [150]. During training, 128 coarse correspondences are sampled with a truncated patch size of $K = 64$ for 3DMatch (3DLoMatch) and 128 and 32 for KITTI, respectively.

Baselines. FOTReg was compared with several learning-based state-of-the-art methods, including FCGF [18], D3Feat [220], SpinNet [221], Predator [14], YOHO [222], CoFiNet [150], and GeoTransformer [9].

5.3.2 Evaluation on 3DMatch and 3DLoMatch.

Datasets. The 3DMatch [31] and 3DLoMatch [14] datasets are commonly used for indoor scenes and contain partial overlapping scene pairs of over 30% and 10% to 30%, respectively.

There are 62 scenes in the 3DMatch dataset, of which 46 are allocated for training, 8 for validation, and 8 for testing. The testing subset comprises 1,623 fragments of point clouds that have partial overlaps, along with their corresponding transformation matrices. The evaluation is performed on the 3DMatch and 3DLoMatch using training data processed by [14]. The point clouds are first downsampled using voxels that have a size of 2.5cm and then different feature descriptors are extracted. Following [14], the values of r_o, r_p, and r_n are set at 3.75cm, 3.75cm, and 10.0cm, respectively.

Metrics. In accordance with Predator [14] and CoFiNet [150], this section assesses performance using three metrics: (1) Inlier Ratio (IR), which represents the proportion of putative correspondences whose residuals are less than a specified threshold (i.e., 0.1m) below the ground-truth transformation, (2) Feature Matching Recall (FMR), which indicates the proportion of pairs of point clouds that have an inlier rate exceeding a defined threshold (i.e., 5%), and (3) Registration Recall (RR), which measures the percentage of point cloud pairs that have a transformation error below a specified threshold (i.e., $RMSE < 0.2m$).

Inlier Ratio and Feature Matching Recall. As the main contribution of FOTReg is that FOTReg jointly adopts the pointwise and structural matchings to estimate the more correct correspondences, this experiment first checks the Inlier Ratio of FOTReg, which is directly related to the quality of extracted correspondences. According to [9], [14], [150], the performance results with varying numbers of correspondences are presented. As shown in Table 5.1 (Top), in terms of Inlier Ratio, FOTReg outperforms all previous methods on both benchmarks, demonstrating remarkable accuracy improvement. Specifically, FOTReg surpasses the second-best baseline, GeoTransformer, by a range of 1.8% to 13.5% on 3DMatch and 1.6% to 10.7% on 3DLoMatch when the sample number varies from 250 to 5000, respectively. Additionally, the decrease in Inlier Ratio with fewer correspondences suggests that the learned scores are well-calibrated, indicating that higher confidence scores correspond to more reliable correspondences. For Feature Matching Recall, as shown in Table 5.1 (Middle), FOTReg achieves the best results. Particularly on 3DLoMatch, which poses greater challenges due to low overlap scenarios, the proposed method shows improvements of at least 0.9%, demonstrating its effectiveness in such cases. The main difference between FOTReg and other baselines is the correspondence prediction strategy, which considers both pointwise and structural matchings and incorporates overlap scores. On the other hand, these baseline methods only consider pointwise matching based on feature similarity, which is insufficient to differentiate features in repetitive regions.

Registration Recall. Registration Recall reflects the final performance on point cloud registration. To evaluate the registration performance, this experiment compares the **RR** obtained

Table 5.1: Results on 3DMatch and 3DLoMatch datasets under varying sample numbers.

	3DMatch					3DLoMatch				
# Samples	5000	2500	1000	500	250	5000	2500	1000	500	250
Method	Inlier Ratio (%) ↑									
FCGF[18]	56.8	54.1	48.7	42.5	34.1	21.4	20.0	17.2	14.8	11.6
D3Feat[220]	39.0	38.8	40.4	41.5	41.8	13.2	13.1	14.0	14.6	15.0
SpinNet [221]	47.5	44.7	39.4	33.9	27.6	20.5	19.0	16.3	13.8	11.1
Predator [14]	58.0	58.4	57.1	54.1	49.3	26.7	28.1	28.3	27.5	25.8
CoFiNet[150]	49.8	51.2	51.9	52.2	52.2	24.4	25.9	26.7	26.8	26.9
YOHO [222]	64.4	60.7	55.7	46.4	41.2	25.9	23.3	22.6	18.2	15.0
GeoTransformer[9]	71.9	75.2	76.0	82.2	85.1	43.5	45.3	46.2	52.9	57.7
FOTReg	**85.4**	**85.7**	**86.1**	**86.4**	**86.9**	**54.2**	**55.1**	**56.3**	**57.7**	**59.3**
	Feature Matching Recall (%) ↑									
FCGF[18]	97.4	97.3	97.0	96.7	96.6	76.6	75.4	74.2	71.7	67.3
D3Feat [220]	95.6	95.4	94.5	94.1	93.1	67.3	66.7	67.0	66.7	66.5
SpinNet [221]	97.6	97.2	96.8	95.5	94.3	75.3	74.9	72.5	70.0	63.6
Predator[14]	96.6	96.6	96.5	96.3	96.5	78.6	77.4	76.3	75.7	75.3
CoFiNet[150]	98.1	98.3	98.1	98.2	98.3	83.1	83.5	83.3	83.1	82.6
YOHO[222]	98.2	97.6	97.5	97.7	96.0	79.4	78.1	76.3	73.8	69.1
GeoTransformer[9]	97.9	97.9	97.9	97.9	97.6	88.3	88.6	88.8	88.6	88.3
FOTReg	**98.5**	**98.6**	**98.5**	**98.6**	**98.6**	**89.5**	**89.7**	**89.7**	**89.6**	**89.4**
	Registration Recall (%) ↑									
FCGF[18]	85.1	84.7	83.3	81.6	71.4	40.1	41.7	38.2	35.4	26.8
D3Feat[220]	81.6	84.5	83.4	82.4	77.9	37.2	42.7	46.9	43.8	39.1
SpinNet[221]	88.6	86.6	85.5	83.5	70.2	59.8	54.9	48.3	39.8	26.8
Predator[14]	89.0	89.9	90.6	88.5	86.6	59.8	61.2	62.4	60.8	58.1
CoFiNet[150]	89.3	88.9	88.4	87.4	87.0	67.5	66.2	64.2	63.1	61.0
YOHO [222]	90.8	90.3	89.1	88.6	84.5	65.2	65.5	63.2	56.5	48.0
GeoTransformer[9]	92.0	91.8	91.8	91.4	91.2	75.0	74.8	74.2	74.1	73.5
FOTReg	**93.1**	**92.8**	**92.9**	**92.5**	**92.4**	**80.9**	**80.4**	**79.7**	**76.0**	**74.6**

by RANSAC in Table 5.1 (bottom), and the proposed method outperforms all the other models with a various number of sampling points on both two datasets. Specifically, FOTReg achieved 93.1% and 80.9% Registration Recall, exceeding the previous best, GeoTransformer,(92.0% RR on 3DMatch) by 1.1% and (80.9% RR on 3DLoMatch) by 5.9%, indicating that FOTReg is effective in situations with both high and low overlap. It demonstrates that incorporating

both pointwise and structural matchings with overlap scores into the correspondence prediction process can alleviate the ambiguity issue. Thus, it obtains better performance than the counterparts that only consider pointwise matching. Figures 5.4 and 5.5 show visual comparison examples on 3DMatch and 3DLoMatch, respectively. It can be easily seen that the proposed method can achieve better results in challenging indoor scenes with a low overlap ratio.

The comparison of the registration results using weighted SVD over correspondences is presented in Table 5.2. Several of the baselines exhibit subpar results or suffer from significant performance degradation. In contrast, FOTReg incorporating weighted SVD attains a registration recall of 87.2% and 60.7% for 3DMatch and 3DLoMatch, respectively. Achieving successful registration necessitates a high inlier ratio when outlier filtering is absent using RANSAC. Nonetheless, as observed in [14], a high inlier ratio does not always guarantee a high registration recall. But, a high inlier ratio can reduce the computation time to calculate the transformation, especially for RANSAC-based methods.

Table 5.2 further counts the average inference time of FOTReg and compares it with that of the baselines. Notably, all methods consist of two stages: extracting dense features or the correspondences and then recovering the transformation using RANSAC or SVD. Table 5.2 reports inference times of the two stages, respectively. Although the proposed method is slightly slower than some baselines in the correspondence prediction stage, it performs well by extracting reliable correspondences.

5.3.3 Evaluation on KITTI

Datasets. In order to perform fair comparisons, the same data splitting as in [12], [18] is adopted for the 11 sequences of LiDAR-scanned outdoor driving scenarios in KITTI. Sequences 0-5 are used for training, 6-7 for validation, and 8-10 for testing. According to [12], the ground truth poses undergo refinement through ICP, and for the purpose of evaluation, only point cloud pairs that have a distance of no more than 10m are employed. The downsampling of the point clouds is performed with a voxel size of 30cm as suggested in [14]. Thresholds are set at $r_o = 45$cm, $r_p = 21$cm, and $r_n = 75$cm.

Metrics. The efficacy of the registration algorithm proposed is assessed through three measures: *Registration Recall* (RR), *Rotation Error* (RRE), and *Translation Error* (RTE). This evaluation is based on the methods used in Predator [14] and CoFiNet [150]. RR is defined as the percentage of successful alignments where the rotation error and translation error are below

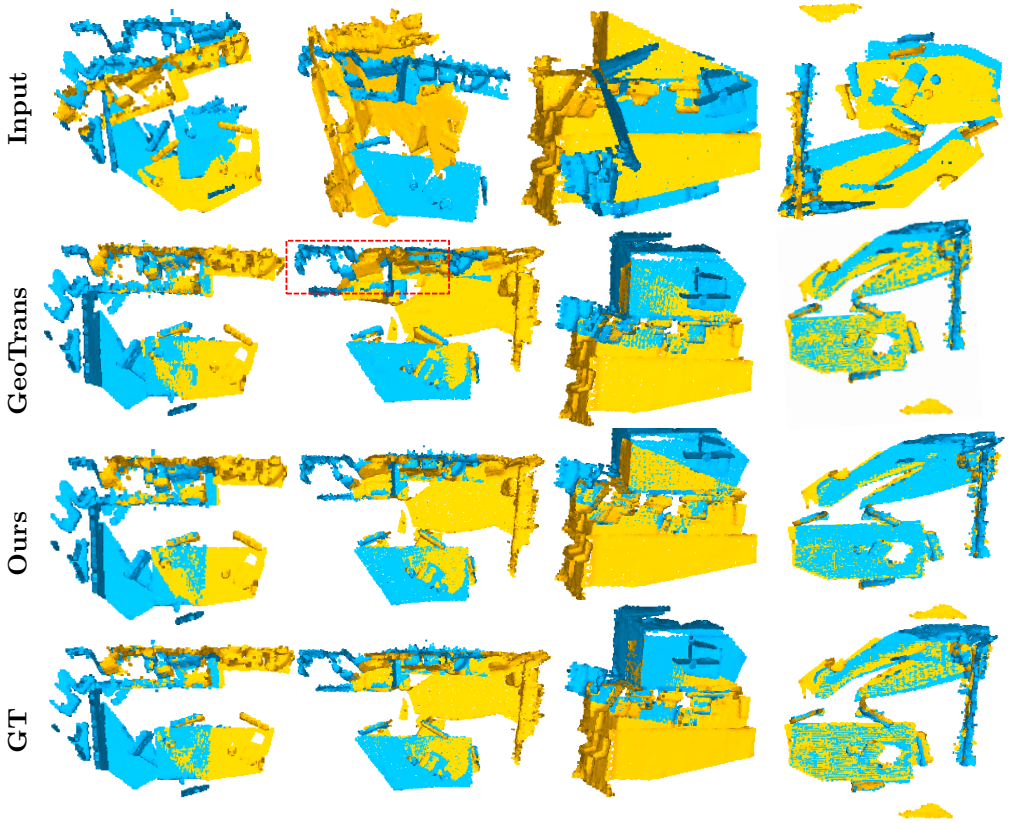


Figure 5.4: Example qualitative registration results for 3DMatch. The unsuccessful cases are enclosed in red boxes.

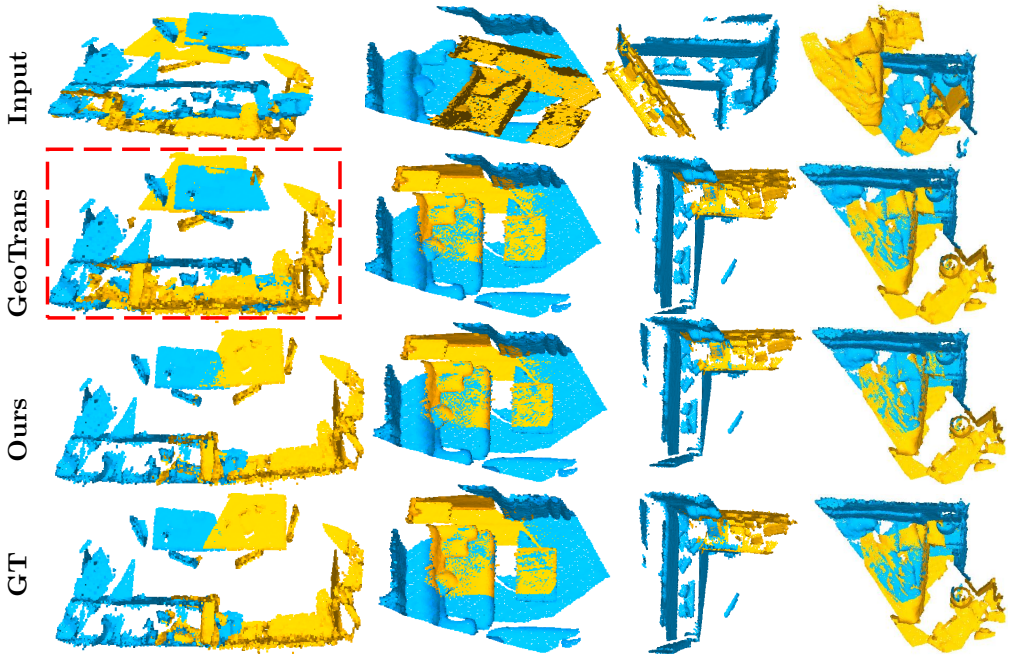


Figure 5.5: Example qualitative registration results for 3DLoMatch. The unsuccessful cases are enclosed in red boxes.

Table 5.2: Results on both 3DMatch and 3DLoMatch datasets under varying sample numbers.

Method	Estimator	Samples	RR		Time (s)		
			3DM	3DLM	Model	Pose	Total
FCGF[18]	RANSAC-50k	5000	85.1	40.1	0.052	3.326	3.378
D3Feat[220]	RANSAC-50k	5000	81.6	37.2	0.024	3.088	3.112
SpinNet [221]	RANSAC-50k	5000	88.6	59.8	60.248	0.388	60.636
Predator [14]	RANSAC-50k	5000	89.0	59.8	0.032	5.120	5.152
CoFiNet[150]	RANSAC-50k	5000	89.3	67.5	0.115	1.807	1.922
GeoTrans[9]	RANSAC-50k	5000	92.0	75.0	**0.075**	**1.558**	**1.633**
FOTReg (Ours)	RANSAC-50k	5000	**93.1**	**80.9**	0.652	1.463	2.115
FCGF[18]	weighted SVD	250	42.1	3.9	0.052	0.008	0.056
D3Feat[220]	weighted SVD	250	37.4	2.8	0.024	0.008	0.032
SpinNet [221]	weighted SVD	250	34.0	2.5	60.248	0.006	60.254
Predator [14]	weighted SVD	250	50.0	6.4	0.032	0.009	0.041
CoFiNet [150]	weighted SVD	250	64.6	21.6	0.115	0.003	0.118
GeoTrans[9]	weighted SVD	250	86.5	59.9	**0.075**	**0.003**	**0.078**
FOTReg (Ours)	weighted SVD	250	**87.2**	**60.7**	0.652	0.002	0.654

set thresholds (i.e., RRE $< 5^\circ$ and RTE $< 2m$). RRE and RTE are respectively calculated as

$$\begin{aligned} RRE &= \arccos \frac{\mathbf{Tr}\left(\boldsymbol{R}^\top \boldsymbol{R}^\star\right) - 1}{2}, \\ RTE &= |\boldsymbol{t} - \boldsymbol{t}^\star|_2, \end{aligned} \tag{5.36}$$

where $\boldsymbol{R}^\star$ and $\boldsymbol{t}^\star$ represent the ground-truth rotation matrix and translation vector, respectively.

Registration results. The comparison to state-of-the-art RANSAC-based methods, including FCGF [18], D3Feat [220], SpinNet [221], Predator [14], CoFiNet [150], and GeoTransformer [9], is conducted. As shown in Table 5.3, the model still obtains the best performance in terms of registration recall and the lowest average RTE and RRE. This verifies the effectiveness of considering both pointwise and structural matchings to generate correspondences.

5.3.4 Generalization on Cross-source Dataset

The generalization ability of learning-based registration algorithms is highly required when the point cloud is acquired from different sensors. To validate the generalizability of the proposed model, this section experiments on the Cross Source Dataset (3DCSR) [22].

Table 5.3: Results on KITTI dataset. Best performance is highlighted in bold.

Method	Estimator	RTE (cm) ↓	RRE (°) ↓	RR(%) ↑
FCGF [18]	RANSAC	9.5	0.30	96.6
D3Feat [220]	RANSAC	7.2	0.30	**99.8**
SpinNet [221]	RANSAC	9.9	0.47	99.1
Predator [14]	RANSAC	6.8	0.27	**99.8**
CoFiNet [150]	RANSAC	8.5	0.41	**99.8**
GeoTrans [9]	RANSAC	7.4	0.27	**99.8**
FOTReg (Ours)	RANSAC	**4.9**	**0.22**	**99.8**

3DCSR. 3DCSR is a challenging dataset for registration due to a mixture of noise, outliers, density differences, partial overlap, and scale variation. This dataset contains two folders: Kinect Lidar and Kinect SFM. Kinect lidar includes 19 scenes from both the Kinect and Lidar sensors, where each scene is cropped into different parts. Kinect SFM consists of 2 scenes from both Kinect and RGB sensors. The RGB images have already been constructed into a point cloud by using the VSFM software. This experiment uses the model trained on 3DMatch since the cross-source dataset is captured in an indoor environment. *RR* represents the proportion of aligned data sets that meet predetermined thresholds for both rotation and translation errors (i.e., RRE $< 15°$ and RTE $< 6m$), indicating a successful alignment.

Table 5.4: The results of the registration on Cross Source Datasets are presented, with the best performance being emphasized in bold.

Method	Estimator	RRE (°) ↓	RTE (cm) ↓	RR(%) ↑
FCGF [18]	RANSAC	7.47	**0.21**	49.6
D3Feat [220]	RANSAC	6.41	0.26	52.0
SpinNet [221]	RANSAC	6.56	0.24	53.5
Predator [14]	RANSAC	6.26	0.27	54.6
CoFiNet [150]	RANSAC	5.76	0.26	57.3
GeoTrans [9]	RANSAC	5.60	0.24	60.2
FOTReg	RANSAC	**5.49**	**0.21**	**63.4**

Registration results. FCGF [18], D3Feat [220], SpinNet [221], Predator [14], CoFiNet [150], and GeoTransformer [9] are chosen as the baselines. Table 5.4 shows that the proposed method obtains the highest accuracies in generalizing the registration ability to the real-world cross-source dataset. Specifically, it outperforms the second-best, GeoTransformer, by more than

3.2% in registration recall (63.4% vs. 60.2%). However, the recall is not high enough, showing that registration challenges on 3DCSR remain.

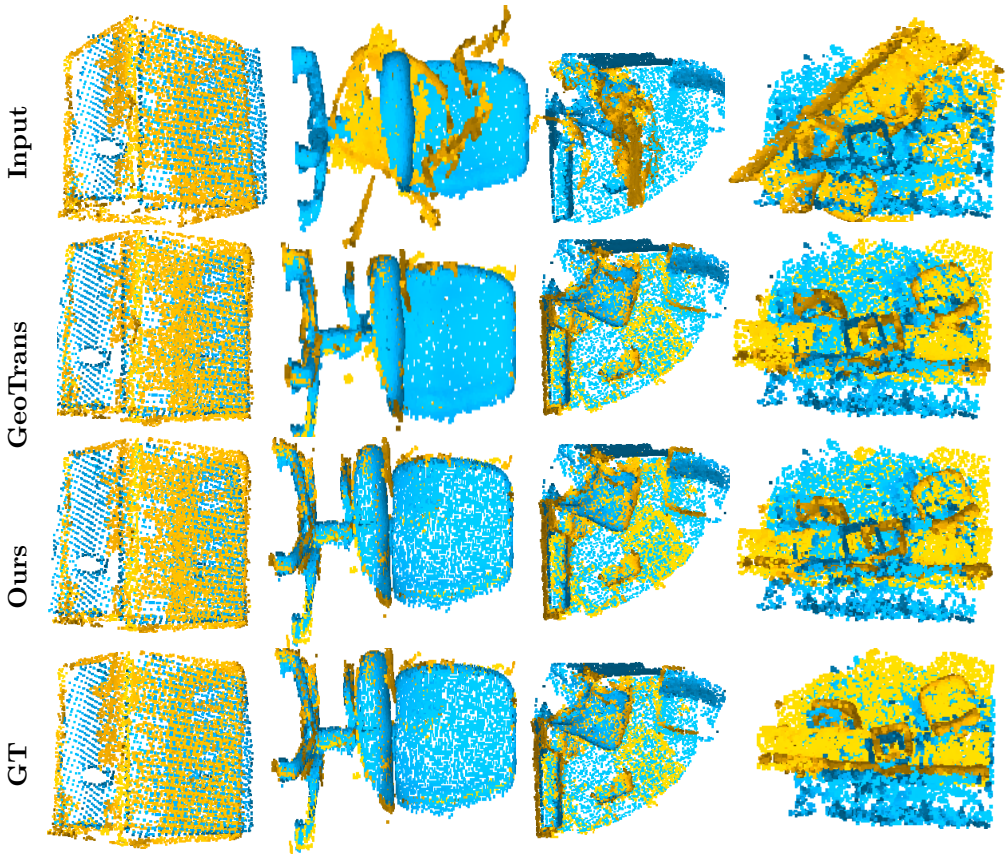


Figure 5.6: Qualitative registration results on cross source dataset. GeoTrans indicates GeoTranformer.

5.3.5 Ablation study

To fully understand FOTReg, an ablation study is conducted on 3DMatch and 3DLoMatch to investigate the contribution of each part. First, the overlap scores are replaced with a uniform distribution, i.e., treating the points in overlap and non-overlap regions equally, to evaluate the effectiveness of overlap scores. As shown in Table 5.5, on 3DMatch, the learned overlap scores improve the performance by nearly 2.0% (92.9% vs. 90.9%) RR, 0.7% (98.5% vs. 97.8%) FMR, and 7.8% (86.1% vs. 68.3%) IR, respectively. Structure matching can boost RR by 1.1% (92.9% vs. 91.8%), FMR by 0.5% (98.5% vs. 98.0%) and IR by 10.2% (86.1% vs. 75.9%), respectively. It also indicates that FOTReg benefits from the overlap scores and structure matching. Table 5.5 also shows that the positional encoding can improve the performance in terms of RR, FMR and IR. On 3DLoMatch, the same results can be concluded.

Table 5.5: An ablation study of individual modules with 1000 samples was conducted. PM stands for point matching and SM represents structure matching. OS represents a point with overlap scores and PE stands for positional embedding.

				3DMatch			3DLoMatch		
PE	OS	PM	SM	RR	FMR	IR	RR	FMR	IR
✓	✓	✓	✓	**92.9**	**98.5**	**86.1**	**79.7**	**89.7**	**55.1**
✓	✓	✓		91.8	98.0	75.9	74.6	88.9	46.4
	✓	✓	✓	90.9	97.8	68.3	67.2	85.6	35.4
	✓	✓		90.2	97.6	63.4	66.1	84.7	33.5
		✓	✓	89.6	97.6	62.9	65.0	84.3	32.1
		✓		88.9	97.5	59.8	64.8	84.0	30.8

5.4 Summary and conclusions

The method proposed in this chapter aims to enhance the accuracy of putative correspondences in point cloud registration. It combines overlap scores, pointwise matchings, and structural matchings in a joint model based on fused Gromov-Wasserstein distance, and adopts a coarse-to-fine approach to estimate correspondences to reduce the burden of GPU memory. Structural matching takes into account both Euclidean and feature differences. The experimental results on a variety of indoor and outdoor, synthetic, and cross-source point clouds have demonstrated the efficacy of the proposed method in improving the accuracy of 3D point cloud registration.

However, the performance of the proposed method is limited in the case of point clouds with density variations, as seen in the Cross-source Dataset 3DCSR. Thus, developing an advanced registration algorithm to address density differences remains a challenging issue.

Chapter 6

Overlap-guided Gaussian Mixture Models for Point Cloud Registration

6.1 Introduction

Chapter 5 introduced a correspondence-based registration framework by simultaneously considering point-wise and structural matchings to estimate correspondences in the overlap regions. It achieved outstanding performance on many benchmarks. However, point-level correspondence-based registration approaches often do not work well under conditions involving varying point densities or repetitive patterns [23]. This limitation is particularly prevalent in indoor environments where low-textured regions or repetitive patterns often occupy a large part of the view. To address this, a probabilistic registration approach is presented in this chapter to mitigate the limitation.

Probabilistic registration methods typically use Gaussian Mixture Models (GMMs) to represent the distribution of point clouds as a density function. Alignment is then achieved through either a correlation-based pipeline or an EM-based optimization pipeline, as described in studies such as [16], [71]. Commonly used point cloud registration formulations, such as CPD and FilterReg, utilize a GMM distribution to represent the geometry of the target point cloud in 3D Euclidean space. The point cloud originating from the source is then fitted to the GMM distribution using the maximum likelihood estimation (MLE) approach. A different probabilistic approach involves methods such as GMMReg [70], and JRMPC [73], which built Gaussian mixture model probability distributions on both the source and target point clouds. Traditional GMM-based registration process that relies on probability often demands considerable computational resources and may take a long time to complete. This can be particularly problematic, especially when managing vast or intricate datasets. To address this issue, it is crucial

to fine-tune algorithms and employ strategies like parallel processing to enhance the speed of registration. This can be a major challenge, particularly when dealing with large or complex datasets. To overcome this, DeepGMR [16] proposed an end-to-end method, which applies a network to learn the parameters of GMM. These probabilistic registration techniques showed greater robustness to noise and density variations than their point-to-point counterpart [223]. However, they typically require their inputs to share the same distribution parameters (e.g., Gaussian Mixture Models). Due to this, they can only handle complete-to-complete [16] or partial-to-complete [223] point cloud registration setups. Partial-to-partial setups commonly encountered in practical scenarios often feature distribution parameters that are not connected. Consequently, relying on cutting-edge methods in such setups may result in subpar performance.

With its built-in permutation invariance and robust capability in learning global features, 3D Transformers prove to be a great tool for point cloud analysis and processing. They have even outperformed non-Transformer algorithms currently available in the community. The field of point cloud learning has seen the emergence of several Transformer-based approaches, with A-SCN [224] being the pioneering example. Later, PCT [225] is a pure global Transformer network that formulates its positional embedding using 3D coordinates of points. Additionally, it incorporates a transformer with an offset attention module, designed to enhance the features of points in their local neighborhood. PointTransformer [98] to construct self-attention networks for general 3D tasks with nearest neighbor search. However, they suffer from the fact that as the size of the feature map increases, the computing and memory overheads of the original Transformer increase quadratically. The primary focus of efforts to decrease the quadratic complexity of attention has been centered on self-attention. For instance, PatchFormer [99] reduces the size of the attention map by first splitting the original point cloud into small patches, and then aggregating the local feature local features within each patch to generate an attention matrix. FastPointTransformer proposes centroid-aware voxelization and devoxelization techniques to reduce space complexity. Nevertheless, these works are not appropriate for feature matching, which requires performing self-attention and cross-attention on features within and between point clouds, respectively.

This chapter proposes an overlap-guided GMM-based registration method, named OGMM, to mitigate the limitations of partial-to-partial setups without using exact point-level correspondences. The registration challenge of aligning point cloud pairs is reformulated as the adjustment of two Gaussian mixtures, achieved by minimizing a statistical dissimilarity measure between the paired mixtures. To measure the likelihood of points being located in the overlapping regions between the source and target point clouds, an overlap score is introduced. This score is obtained using a Transformer-based deep neural network. The input point cloud is represented using GMMs, guided by the overlap score, to effectively capture the overlapping ar-

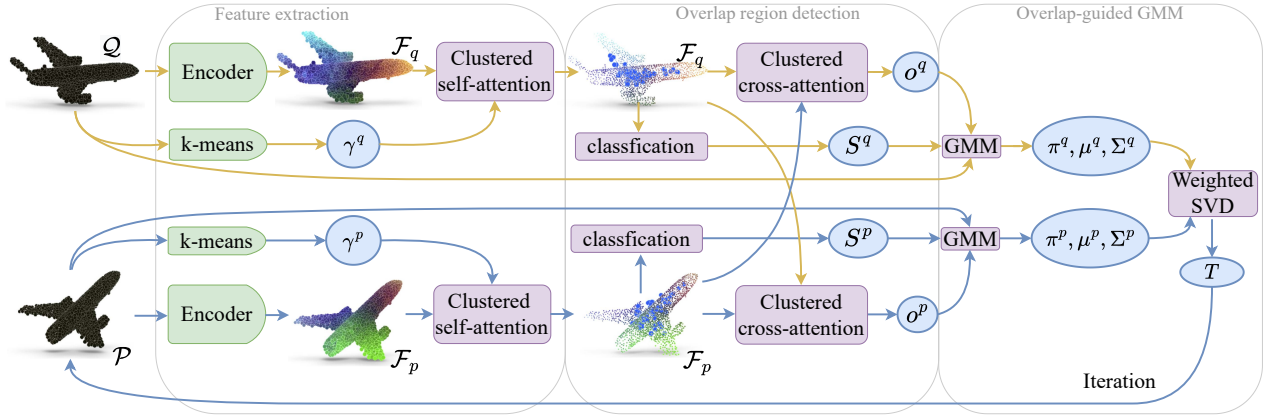


Figure 6.1: The proposed OGMM consists of three modules: feature extraction, overlap region detection, and overlap-guided GMM for registration. The shared weighted encoder extracts point-level features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ from point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. The self-attention module updates the point-wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$. The overlap region detection module projects the updated features $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ to overlap scores $\boldsymbol{o}_p, \boldsymbol{o}_q$, respectively. $\boldsymbol{\mathcal{F}}_p$, $\boldsymbol{\mathcal{F}}_q$, $\boldsymbol{o}_p$ and $\boldsymbol{o}_q$ are used to estimate GMMs of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. The weighted SVD estimates the rigid transformation T based on the estimated distributions.

eas. However, the computational and memory requirements of self-attention or cross-attention in Transformer networks scale quadratically with the size of point clouds (N^2), limiting their practicality when dealing with large-scale point cloud datasets. Therefore, this chapter introduces the idea of *clustered attention*, which is a fast approximation of self-attention. Clustered attention groups a set of points into J clusters and compute the attention for these clusters only, making the complexity linear with the number of clusters, i.e., $N \cdot J$, where $J << N$. This proposed method is inspired by DeepGMR [16], but it differs from it in two ways. First, the probabilistic approach can tackle partially overlapping point cloud registration challenges through the implementation of an overlap score restriction. Second, this proposed method applies a network to learn a consistent GMM representation across feature and geometric space rather than fitting a GMM in a single feature space.

6.2 Methodology

Rigid point cloud registration seeks to determine the optimal transformation matrix $T \in SE(3)$, which consists of a rotation $R \in SO(3)$ and a translation $\boldsymbol{t} \in \mathbb{R}^3$, that aligns the source point cloud $\boldsymbol{\mathcal{P}} = \{\boldsymbol{p}_i \in \mathbb{R}^3 \big| i = 1, 2, ..., N\}$ with the target point cloud $\boldsymbol{\mathcal{Q}} = \{\boldsymbol{q}_j \in \mathbb{R}^3 \big| j = 1, 2, ..., M\}$. Here, N and M represent the number of points in $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. Fig. 6.1 illustrates the proposed framework that consists of three modules: feature extraction, overlap region detection, and overlap-guided GMM for registration. The shared weighted encoder first extracts point-

wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ from point clouds $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, respectively. The clustered self-attention module then updates the point-wise features $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ to capture global context. Next, the overlap region detection module projects the updated features $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ to overlap scores $\boldsymbol{o}_p, \boldsymbol{o}_q$, respectively. $\boldsymbol{\mathcal{F}}_p$, $\boldsymbol{\mathcal{F}}_q$, $\boldsymbol{o}_p$ and $\boldsymbol{o}_q$ are then used to estimate the distributions (GMMs) of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. Finally, weighted SVD is adopted to estimate the rigid transformation T based on the estimated distributions.

6.2.1 Feature extraction

The feature extraction network consists of a Dynamic Graph Convolutional Neural Network (DGCNN), positional encoding, and a clustered self-attention network. Given a point cloud pair $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, DGCNN extracts their associated features $\boldsymbol{\mathcal{F}}_p = \{\boldsymbol{f}_{p_i} \in \mathbb{R}^d \big| i = 1, 2, ..., N\}$ and $\boldsymbol{\mathcal{F}}_q = \{\boldsymbol{f}_{q_j} \in \mathbb{R}^d \big| j = 1, 2, ..., M\}$. Here, $d = 512$.

6.2.2 Attention module

Transformer training and inference in previous works can be computationally expensive owing to their self-attention mechanism having a quadratic complexity for long sequences of representations, particularly in high-resolution correspondence prediction tasks. To mitigate this limitation, the proposed cluster-based Transformer architecture works after local feature extraction. The features, $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$, are processed through the attention module to extract context-dependent features for each point. This self-attention module transforms the DGCNN features into more meaningful representations to improve the matching process.

Spherical positional encoding. Transformers are typically fed with only high-level features, which lack the explicit encoding of a point cloud [10], [14]. This leads to less discriminative features, causing numerous outlier matches and severe matching ambiguity, especially in cases with low overlap [9]. A common solution is to add the positional encoding of 3D point coordinates, which assigns intrinsic geometric properties to the per-point feature by adding unique positional information that enhances distinctions among point features in indistinctive regions [98]. However, the coordinate-based attentions generated as a result are not invariant to transformations, as pointed out in the study [9]. This leads to a problem since the input point clouds could be in any arbitrary pose, and registration requires transformation invariance. To address this issue, this section designs spherical positional encoding, which leverages the distances and angles calculated between the points, to embed transformation-invariant geometric information of the points. Specifically, given a point $\boldsymbol{p}_i \in \boldsymbol{\mathcal{P}}$, it selects the $k > 0$ nearest neighbors $\mathcal{K}_i$ of $\boldsymbol{p}_i$ and compute the centroid $\boldsymbol{p}_c = \sum_{i=1}^{N} \boldsymbol{p}_i$ of mass of $\boldsymbol{\mathcal{P}}$. For each $\boldsymbol{p}_x \in \mathcal{K}_i$, this module first denotes the angle between the vectors $\boldsymbol{p}_i - \boldsymbol{p}_c$ and $\boldsymbol{p}_x - \boldsymbol{p}_c$ as α_{ix}. The positional encoding

$\boldsymbol{f}_{p_i}^{pos} \in \mathbb{R}^d$ of $\boldsymbol{p}_i$ is as

$$\boldsymbol{f}_{p_i}^{pos} = \varphi\left(\|\boldsymbol{p}_i - \boldsymbol{p}_c\|_2\right) + \max_{x \in \mathcal{K}_i}\{\phi\left(\alpha_{ix}\right)\}, \tag{6.1}$$

where φ and ϕ are two MLPs, and each MLP consists of a linear layer and one ReLU nonlinearity function [226]. The proposed positional encoding is invariant to rigid transformation since the distance and angles are invariant to transformation. Then, features $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}}$ of $\boldsymbol{\mathcal{P}}$ are updated by $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}} = \{\boldsymbol{f}_{\boldsymbol{p}_i}^{pos} + \boldsymbol{f}_{\boldsymbol{p}_i}\}$. The same operation is also applied in $\boldsymbol{\mathcal{Q}}$.

Cluster-based self-attention. Although, the extracted local features that provides valuable local information, have a limited receptive field that could potentially compromise the distinctiveness of regions. As a result, these features may not provide a comprehensive understanding of the relationships between different regions. On the other hand, humans are able to find correspondences between these regions not only through the analysis of local neighborhood structures but also by utilizing context. By incorporating the surrounding context into their interpretation, humans are able to achieve a more comprehensive understanding of the relationships between regions, leading to a more accurate representation of the scene. Self-attention is thus introduced to model global structures by establishing long-range dependencies. Standard attention, which requires significant memory usage, is a computationally intensive process. This module exploits this idea to improve the computational complexity of self-attention. Specifically, Wasserstein K-Means [227] is used to cluster point cloud $\boldsymbol{\mathcal{P}}$ (or $\boldsymbol{\mathcal{Q}}$) into J non-overlapping clusters in geometric space by $\boldsymbol{\gamma}^p \in \{0,1\}^{N \times J}$ such that, $\gamma_{ij}^p = 1$ (or $\gamma_{ij}^q = 1$), if the i-th point of $\boldsymbol{\mathcal{P}}$ (or $\boldsymbol{\mathcal{Q}}$) belongs to the j-th cluster (denoted as $\bar{\boldsymbol{p}}_j$ or $\bar{\boldsymbol{q}}_j$) and 0 otherwise. The clustered attention can be computed by using this partitioning. First, the cluster centroids $\boldsymbol{\mathcal{F}}_{\bar{p}} = \{\boldsymbol{f}_{\bar{p}_j}\}_{i=1}^J$ and $\boldsymbol{\mathcal{F}}_{\bar{q}} = \{\boldsymbol{f}_{\bar{q}_j}\}_{i=1}^J$ of the points in each of these J clusters in feature space are calculated as follows,

$$\begin{aligned} \boldsymbol{f}_{\bar{p}_j} &= \frac{1}{\sum_k^N \gamma_{kj}^p} \sum_{i=1}^N \gamma_{ij}^p \boldsymbol{f}_{p_i}, \\ \boldsymbol{f}_{\bar{q}_j} &= \frac{1}{\sum_k^M \gamma_{kj}^q} \sum_{i=1}^M \gamma_{ij}^q \boldsymbol{f}_{q_j}. \end{aligned} \tag{6.2}$$

A multi-attention layer with four parallel attention heads [10] is then applied to update $\boldsymbol{\mathcal{F}}_{\boldsymbol{p}}$ in parallel via

$$\boldsymbol{f}_{\boldsymbol{p}_i} \leftarrow \boldsymbol{f}_{\boldsymbol{p}_i} + \mathrm{MLP}\left(\sum_{j=1}^J \alpha_{ij}^{\boldsymbol{p}} W_V^s \boldsymbol{f}_{\bar{\boldsymbol{p}}_j}\right), \tag{6.3}$$

where $\alpha_{ij}^{\boldsymbol{p}}$ is the element of matrix $\boldsymbol{\alpha}^{\boldsymbol{p}} = \mathrm{Softmax}(\boldsymbol{S})$ with $\boldsymbol{S} = (W_Q^s \boldsymbol{f}_{\boldsymbol{p}_i})^\top W_K^s \boldsymbol{f}_{\bar{\boldsymbol{p}}_j}$. Here, $W_Q^s \in \mathbb{R}^{N \times d}, W_K^s \in \mathbb{R}^{J \times d}$ and $W_V^s \in \mathbb{R}^{J \times d}$ are the query, key and value matrices. The self-attention features for $\boldsymbol{\mathcal{Q}}$ are updated in the same way. MLP$(\cdot)$ refers to a fully connected network with three layers, featuring instance normalization [228] and ReLU activations [226]

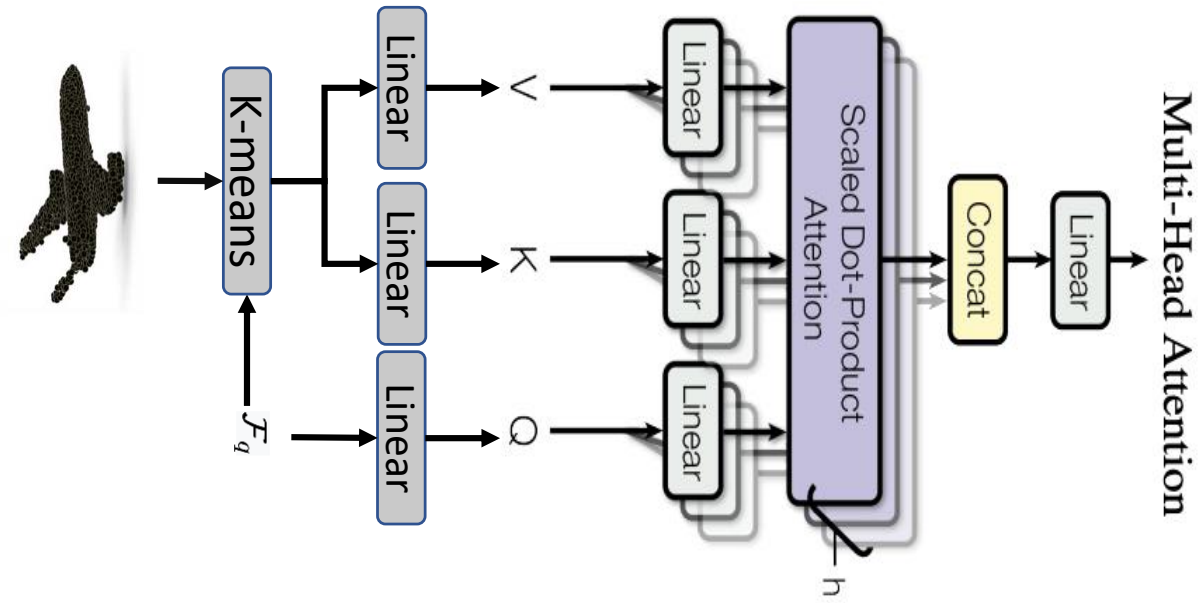


Figure 6.2: The framework of the clustered self-attention.

following the first two layers. The diagram of the proposed clustered self-attention is illustrated in Figure 6.2.

Cluster-based cross-attention. Cross-attention is a typical module for point cloud registration tasks and it accomplishes feature exchange between two input point clouds. This module performs an initial step of updating the cluster centroids $\boldsymbol{f}_{\bar{p}_j}$ and $\boldsymbol{f}_{\bar{q}_j}$ based on the self-attention feature matrices $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ extracted from $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ respectively, according to the formula specified in Eq. (6.2). The transformed features is denoted as $\boldsymbol{\mathcal{F}}_p^t$ and $\boldsymbol{\mathcal{F}}_p^t$ attained by cross-attention via

$$\boldsymbol{f}_{\boldsymbol{p}_i}^t \leftarrow \boldsymbol{f}_{\boldsymbol{p}_i} + \mathrm{MLP}\left(\sum \beta_{ij}^{\boldsymbol{p}} W_V^c \boldsymbol{f}_{\bar{\boldsymbol{q}}_j}\right), \tag{6.4}$$

where $\beta_{ij}^{\boldsymbol{p}}$ is the element of matrix $\boldsymbol{\beta}^{\boldsymbol{p}} = \mathrm{SoftMax}(\boldsymbol{C})$ with $\boldsymbol{C} = (W_Q^c \boldsymbol{f}_{\boldsymbol{p}_i})^\top W_K^c \boldsymbol{f}_{\bar{\boldsymbol{q}}_j}$. Here, $W_Q^c \in \mathbb{R}^{N\times d}, W_K^c \in \mathbb{R}^{J\times d}$ and $W_V^c \in \mathbb{R}^{J\times d}$ are the query, key and value matrices. The notation $\mathrm{MLP}(\cdot)$ represents a fully connected network with three layers, which includes instance normalization [228] and ReLU activations [226] following the first two layers. A cross-attention block is applied in both directions to ensure information flow in both directions, $\boldsymbol{\mathcal{P}} \rightarrow \boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{Q}} \rightarrow \boldsymbol{\mathcal{P}}$. Figure 6.3 shows the diagram of the proposed clustered self-attention.

Overlap score prediction. To deal with non-overlapping points, an overlap score prediction block is proposed. After obtaining the conditioned features $\boldsymbol{\mathcal{F}}_p^t$ and $\boldsymbol{\mathcal{F}}_q^t$, the per-point overlap score $\boldsymbol{o}_{p_i} \in [0, 1]$ can be computed by

$$\begin{aligned} w_{ij} &= \sigma\left({\boldsymbol{f}_{p_i}^t}^\top \boldsymbol{f}_{q_j}^t / \tau\right), \\ \boldsymbol{o}_{p_i} &= g_\beta\left(\mathrm{cat}\left[\boldsymbol{f}_{q_i}^t, \boldsymbol{w}_i^\top g_\alpha\left(\boldsymbol{\mathcal{F}}_q^t\right)\right]\right), \end{aligned} \tag{6.5}$$

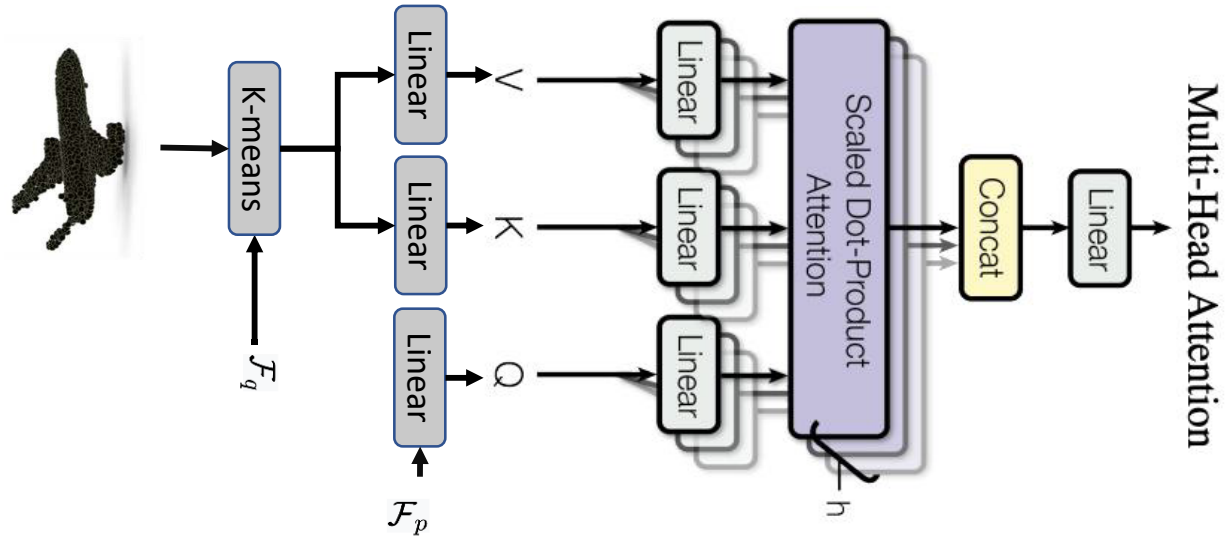


Figure 6.3: The framework of the clustered cross-attention.

where σ is a Softmax function, and $\tau > 0$ is a learned parameter that controls the soft assignment. When $\tau \to 0$, w_{ij} converges to a hard nearest-neighbor assignment. $g_\alpha(\cdot) : \mathbb{R}^d \to [0, 1]$ and $g_\beta(\cdot) : \mathbb{R}^{d+1} \to [0, 1]$ are linear layers followed by an instance normalization layer and a sigmoid activation with different parameters α and β, respectively.

6.2.3 Overlap-guided GMM for registration

The OGMM proposes to represent discrete point clouds by using Gaussian Mixture Models (GMM). The GMM sets up a generative probability distribution in 3D space that is made up of multiple Gaussian densities, each with its own weight, amounting to L total Gaussian densities [16], with the form as

$$p(\boldsymbol{x}) = \sum_{j=1}^{L} \pi_j \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right), \boldsymbol{x} \in \mathbb{R}^3. \tag{6.6}$$

Each Gaussian distribution $\mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\right)$ is appertained to as a component in the Gaussian Mixture Model, and is characterized by its mean $\boldsymbol{\mu}_j$ and covariance $\boldsymbol{\Sigma}_j$. The components are combined through a set of normalizing mixing coefficients $\{\pi_1, \pi_2, \cdots, \pi_L\}$, which represent the prior probability of selecting the j-th component.

Learning posterior. Traditional probabilistic registration often requires significant computational resources and may have a lengthy processing time. To overcome this challenge, inspired by DeepGMR, OGMM employs a network to learn a GMM. Specifically, OGMM first applies a classification head ϕ_θ that takes as input $\boldsymbol{\mathcal{F}}_p$ and $\boldsymbol{\mathcal{F}}_q$ and outputs joint log probabilities, and a softmax operator that acts on log probabilities to generate a probability matrix $\boldsymbol{S}^p$ and $\boldsymbol{S}^q$, respectively. The GMM parameters $\boldsymbol{\Theta}_p$ for point cloud $\boldsymbol{\mathcal{P}}$ consists of L triples $(\pi_j^p, \boldsymbol{\mu}_j^p, \boldsymbol{\Sigma}_j^p)$,

where π_j^p is a scalar mixture weight, $\boldsymbol{\mu}_j^p$ is a 3×1 mean vector and $\boldsymbol{\Sigma}_j^p$ is a 3×3 covariance matrix of the j-th component. For partial overlapping registration tasks, given the outputs $\boldsymbol{S}^p$ of ϕ_θ together with the point coordinates $\boldsymbol{\mathcal{P}}$, and overlap scores $\boldsymbol{o}_p$, the GMM parameters can be written as

$$\begin{aligned} n_p &= \sum_{i=1}^{N} \boldsymbol{o}_{p_i}, \\ \pi_j^p &= \frac{1}{\epsilon + n_p} \sum_{i=1}^{N} \boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p, \\ \boldsymbol{\mu}_j^p &= \frac{1}{\epsilon + n_p \pi_j^p} \sum_{i=1}^{N} \boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p \boldsymbol{p}_i, \\ \boldsymbol{\Sigma}_j^p &= \frac{\sum_{i=1}^{N} \boldsymbol{o}_{p_i} \boldsymbol{s}_{ij}^p \left(\boldsymbol{p}_i - \boldsymbol{\mu}_j^p\right) \left(\boldsymbol{p}_i - \boldsymbol{\mu}_j^p\right)^\top}{\epsilon + n_p \pi_j^p}, \end{aligned} \tag{6.7}$$

where ϵ=$1e{-}4$ is used to avoid zero in the denominator. To deal with outliers, adding a Gaussian kernel density is straightforward. OGMM distinguishes itself from the prevailing GMM-based techniques, as it does not make the conventional assumption of uniform distribution for the outliers. To achieve this, this module first defines $\boldsymbol{s}_{i,L+1}^p$=$1.0{-}\boldsymbol{o}_{p_i}$, then it can compute the additional components

$$\begin{aligned} N_L &= \sum_{k}^{N} \boldsymbol{s}_{k,L+1}^p, \\ \boldsymbol{\mu}_{L+1}^q &= \frac{1}{N_L} \sum_{i=1}^{N} \boldsymbol{s}_{i,L+1}^p \boldsymbol{p}_i, \\ \boldsymbol{\Sigma}_{L+1}^p &= \frac{1}{N_L} \sum_{i=1}^{N} \boldsymbol{s}_{iL+1}^p \left(\boldsymbol{p}_i - \boldsymbol{\mu}_{L+1}^p\right) \left(\boldsymbol{p}_i - \boldsymbol{\mu}_{L+1}^p\right)^\top. \end{aligned} \tag{6.8}$$

By the same operation, it can get the $\pi_j^q, \boldsymbol{\mu}_j^q$ and $\boldsymbol{\Sigma}_j^q$ for target point cloud, when giving $\boldsymbol{S}^q, \boldsymbol{o}_q, n_q$ and $\boldsymbol{\mathcal{Q}}$. The GMMs of point set $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ are then given as

$$\begin{aligned} \mathbf{G}_{\mathcal{P}}(\boldsymbol{x}) &= \frac{n_p}{N} \sum_{i=1}^{L} \pi_j^p \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j^p, \boldsymbol{\Sigma}_j^p\right) + \left(1 - \frac{n_p}{N}\right) \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_{L+1}^p, \boldsymbol{\Sigma}_{L+1}^p\right), \\ \mathbf{G}_{\mathcal{Q}}(\boldsymbol{x}) &= \frac{n_q}{M} \sum_{i=1}^{L} \pi_j^q \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_j^q, \boldsymbol{\Sigma}_j^q\right) + \left(1 - \frac{n_q}{M}\right) \mathcal{N}\left(\boldsymbol{x} | \boldsymbol{\mu}_{L+1}^q, \boldsymbol{\Sigma}_{L+1}^q\right). \end{aligned} \tag{6.9}$$

Similarly, the feature centroids $\{\boldsymbol{\nu}_j^p\}_{j=1}^L$ and $\{\boldsymbol{\nu}_j^q\}_{j=1}^L$ of $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ can be calculated by replacing $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ with $\boldsymbol{f}_{p_i}$ and $\boldsymbol{f}_{q_i}$, respectively.

Estimating the transformation. Given the estimated GMMs parameters estimated through Eq. (6.9) as well as feature centroids $\{\boldsymbol{\nu}_j^p\}_{j=1}^{L+1}$ and $\{\boldsymbol{\nu}_j^q\}_{j=1}^{L+1}$, the cluster-level matching matrix

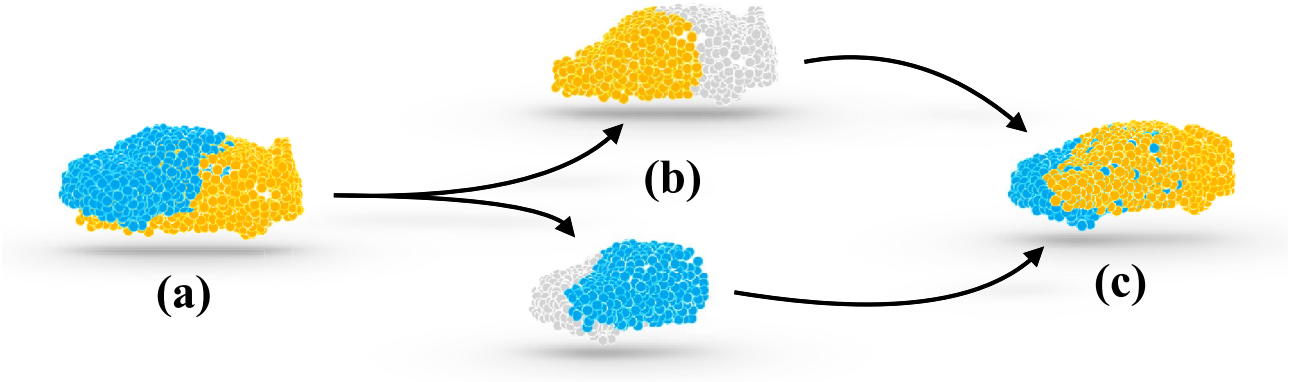


Figure 6.4: Given (a) input partial point clouds, OGMM detects (b) the overlap regions that are then used for the estimation of (c) the rotation and translation that register the input point clouds. The non-overlap regions in (b) are shown in grey. The proposed approach focuses on the geometric information in the overlap regions to perform the point cloud registration.

Γ is first calculated by solving the following optimal transport (OT) problem [136] as

$$\begin{aligned} &\min_{\Gamma} \sum_{i=1}^{L+1}\sum_{j=1}^{L+1} \Gamma_{ij}\|\boldsymbol{\nu}_i^p - \boldsymbol{\nu}_j^q\|_2^2, \\ &\text{s.t., } \Gamma\mathbf{1}_M = \boldsymbol{\pi}^p, \Gamma^\top\mathbf{1}_N = \boldsymbol{\pi}^q, \Gamma_{ij} \in [0,1], \end{aligned} \tag{6.10}$$

where $\boldsymbol{\pi}^t = (\frac{n_t}{N}\pi_1^t, \frac{n_t}{K}\pi_1^t, \cdots, \frac{n_t}{N}\pi_L^t, 1-\frac{n_p}{K}), t \in \{p,q\}, K{=}M,N$. The minimization of Eq. (6.10) can be solved in polynomial time as a linear program, and this issue can be handled by adopting an efficient version of the Sinkhorn-Knopp algorithm [167]. After obtaining the Γ, the transformation is then can be calculated by

$$\min_{T} L\sum_{i=1}^{L}\sum_{j=1}^{L} \Gamma_{ij}\|T(\boldsymbol{\mu}_i^p) - \boldsymbol{\mu}_j^q\|. \tag{6.11}$$

Finally, the solution for transformation T can be expressed in a closed form by utilizing a weighted version of the SVD solution, as stated in the reference [16].

Fig. 6.4 shows an example of registration where the overlap regions that the proposed approach can automatically determine are colored with the respective point cloud colors (non-overlap regions are in grey). This example depicts a case where the overlap between the two point clouds is 50%.

6.2.4 Training

The feature extractor is trained by jointly optimizing three tasks, finding overlap regions, clustering point clouds, and estimating rigid transformations.

Overlap score loss. The goal of the overlap score loss is to detect the overlap region between $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$. Given ground-truth $\bar{\boldsymbol{T}}$, the ground-truth $\bar{\boldsymbol{o}}_{p_i}$ of $\boldsymbol{p}_i$ is defined

$$\bar{\boldsymbol{o}}_{p_i} = \begin{cases} 1, & \left(\min_{q_j \in \boldsymbol{\mathcal{Q}}} \|\bar{\boldsymbol{T}}(\boldsymbol{p}_i) - \boldsymbol{q}_j\|\right) < \eta \\ 0, & \text{otherwise} \end{cases}, \tag{6.12}$$

$\bar{\boldsymbol{o}}_{q_j}$ is calculated in the same way. The overlap score loss is defined as $\mathcal{L}_O = \frac{1}{2}\left(\mathcal{L}_{\boldsymbol{\mathcal{P}}} + \mathcal{L}_{\boldsymbol{\mathcal{Q}}}\right)$, where

$$\mathcal{L}_{\boldsymbol{\mathcal{P}}} = -\frac{1}{|\boldsymbol{\mathcal{P}}|}\sum_i \left(\bar{\boldsymbol{o}}_{\boldsymbol{p}_i} \log \boldsymbol{o}_{\boldsymbol{p}_i} + (1 - \bar{\boldsymbol{o}}_{\boldsymbol{p}_i}) \log\left(1 - \boldsymbol{o}_{\boldsymbol{p}_i}\right)\right). \tag{6.13}$$

By the same operation, $\mathcal{L}_{\boldsymbol{\mathcal{Q}}}$ can be calculated.

Global registration loss. The training of the model is done using a registration error-based loss function. To effectively handle the partially-overlapping issue, a robust error metric is utilized which minimizes computational overhead. The registration loss is expressed as follows:

$$\mathcal{L}_g = \sum_{\hat{\boldsymbol{p}} \in \hat{\boldsymbol{\mathcal{P}}}} \psi_\nu\left(\mathcal{D}\left(\hat{\boldsymbol{p}}, m\left(\bar{\boldsymbol{T}}(\boldsymbol{p}), \boldsymbol{\mathcal{Q}}\right)\right)\right). \tag{6.14}$$

Here, $m(\boldsymbol{x}, \boldsymbol{\mathcal{Q}})$ maps point $\boldsymbol{x}$ to its nearest point in $\boldsymbol{\mathcal{Q}}$. $\hat{\boldsymbol{\mathcal{P}}}$ denotes the transformed $\boldsymbol{\mathcal{P}}$ using the estimated transformation. $\mathcal{D}(\cdot,\cdot)$ defines as the Euclidean distance of two vectors. ψ_ν is the Welsch's function [229] as $\psi_\nu(x) = 1 - \exp\left(-\frac{x^2}{2\nu^2}\right)$. $\nu > 0$ is a user-specified parameter. Figure 6.5 shows the graphs of ψ_ν with different parameters. Since $\psi_\nu(x)$ is monotonically increasing on $x \in [0, +\infty)$, this formulation penalizes deviation between the point sets. As ψ_ν is upper bounded by 1, it is not sensitive to large deviations caused by outliers and partial overlaps. Moreover, when ν approaches zero, $\mathcal{L}_g\left(\hat{\boldsymbol{\mathcal{P}}}, \boldsymbol{\mathcal{Q}}\right) = \sum_{\hat{\boldsymbol{p}} \in \bar{\boldsymbol{\mathcal{P}}}} \psi_\nu\left(\min_{\boldsymbol{q} \in \boldsymbol{\mathcal{Q}}} \|\bar{\boldsymbol{p}} - \boldsymbol{q}\|_2^2\right) + \sum_{\boldsymbol{q} \in \boldsymbol{\mathcal{Q}}} \psi_\nu\left(\min_{\bar{\boldsymbol{p}} \in \hat{\boldsymbol{\mathcal{P}}}} \|\hat{\boldsymbol{p}} - \boldsymbol{q}\|_2^2\right)$ approaches the l_0-norm of the vector $[\min_{\boldsymbol{q} \in \boldsymbol{\mathcal{Q}}} \|\hat{\boldsymbol{p}}_1 - \boldsymbol{q}\|_2^2, \cdots, \min_{\boldsymbol{q} \in \boldsymbol{\mathcal{Q}}} \|\hat{\boldsymbol{p}}_N - \boldsymbol{q}\|_2^2]$. Therefore, this formulation encourages a reduction in the density of the point-to-point distance within the sets.

Clustering-based loss. To ensure that the network learns a consistent GMM representation across feature and geometric space rather than fit a GMM in a single feature space, this leads to the following loss.

$$\mathcal{L}_c = -\sum_{ij} \gamma_{ij}^p \log \frac{\exp\left(-\mathcal{D}\left(\boldsymbol{p}_i, \boldsymbol{\mu}_j^{\boldsymbol{p}}\right)\right)}{\sum_l \exp\left(-\mathcal{D}\left(\boldsymbol{p}_i, \boldsymbol{\mu}_l^{\boldsymbol{p}}\right)\right)} - \sum_{ij} \gamma_{ij}^q \log \frac{\exp\left(-\mathcal{D}\left(\boldsymbol{q}_i, \boldsymbol{\mu}_j^{\boldsymbol{q}}\right)\right)}{\sum_l \exp\left(-\mathcal{D}\left(\boldsymbol{q}_i, \boldsymbol{\mu}_l^{\boldsymbol{q}}\right)\right)}. \tag{6.15}$$

6.3 Experiments

This section performs extensive experiments and ablation studies on both synthetic and real-world point cloud datasets, including the ModelNet40 [207], 7Scenes [208], and ICL-NUIM

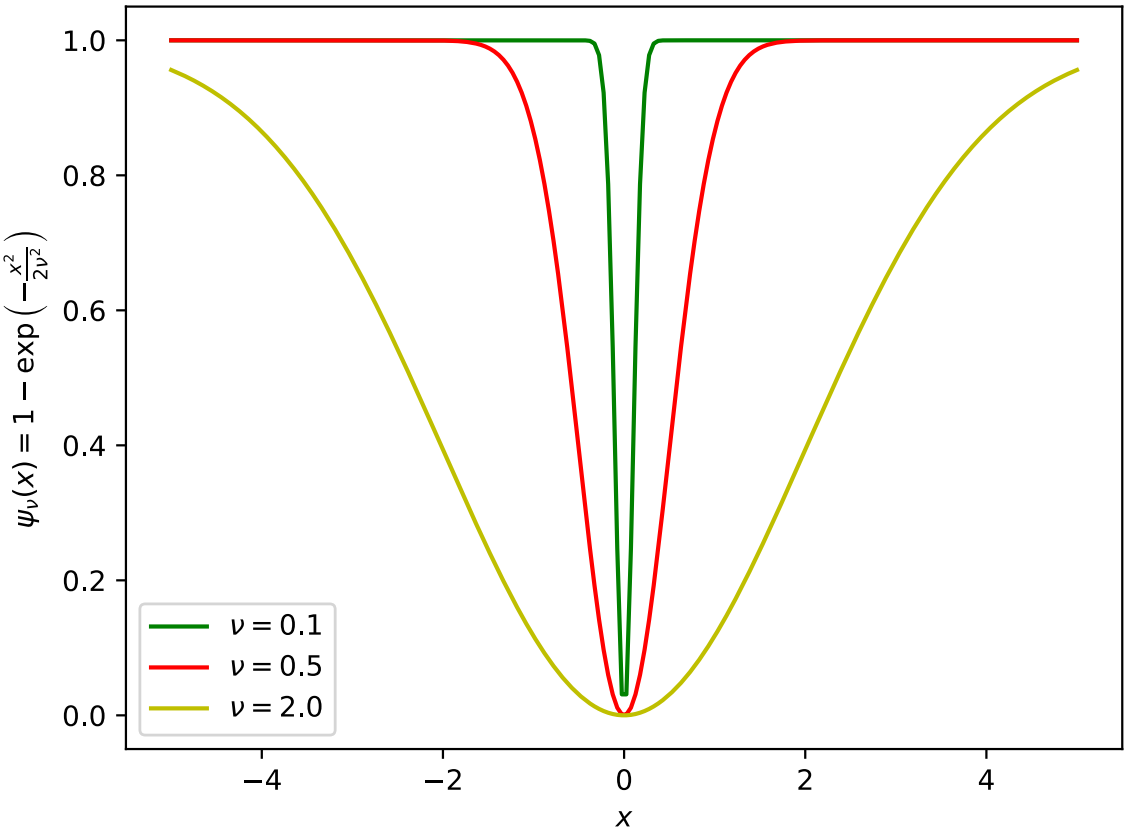


Figure 6.5: The graphs of function $\psi_\nu(x)$ with different values of ν. As ν decreases, the function ψ_ν approaches the l_0 norm.

[230]. Unless otherwise specified, $J = 72, K = 5, \eta = 0.1$. This implementation is built on the PyTorch library. This implementation used AdamW as an optimizer with a base learning rate of 0.001. The batch size is 32, and the learning rate was reduced by a factor of 0.7 every 20 epochs. OGMM was trained for 200 epochs. All of the models were trained on two Tesla V100-PCI-E-32G GPUs. The GMM components are set as $L = 48$. To alleviate the local minima clustering solution, this experiment initializes the centroids of Wasserstein K-Means using the farthest point sampling strategy and the equal partition constraint.

6.3.1 Comparisons

This experiment evaluates the performance of the proposed approach by comparing it to state-of-the-art methods. The proposed method is a learning-based probabilistic registration approach, so the comparison is made with other probabilistic methods, including CPD [72], GMMReg [70], SVR [231], DeepGMR [16], and FilterReg [74]. The performance of each method is measured by running their code on selected datasets. For these traditional methods, this

experiment uses the implementations provided by probreg [232]. This experiment improves the DeepGMR by replacing its encoder of DeepGMR with the encoder used in the OGMM for a fair comparison. This experiment also includes point-level correspondence methods: traditional methods (i.e., ICP [42], FGR [75]), using implementations from Open3D [217], and the learning-based state-of-the-art methods RGM [15], OMNET [123], RIENet [21], and REGTR [134] (using authors' implementation).

6.3.2 Evaluation metrics

This section evaluates the registration quality by using the *Mean Absolute Error (MAE)* between the estimated rotation angle θ_{est} and ground truth θ_{gt}, and the estimated translation t_{est} and ground truth t_{gt} [10], [15]. Rotation metrics are in *degrees*, while translation metrics are in *cm*. *Clip Chamfer Distance (CCD)* [15] metric is also employed to determine the level of alignment between the two point clouds [15]. To counteract the effects of outliers in partial-to-partial registration, any pairs of points with a distance greater than 0.1 are discarded by setting a threshold of $d = 0.1$.

6.3.3 Datasets

ModelNet40 [207] comprises 12,311 meshed CAD models across 40 categories. In accordance with [233], this experiment divides the dataset into two setups: *same-category* and *cross-category*. The same-category setup involves 20 categories for both training and testing. The cross-category setup, on the other hand, consists of 20 categories that are disjoint between the training and testing sets for the evaluation of generalization performance. Following RGM [15], each category follows the official train/test splits. To select models for evaluation, 80% of the official train split is used as the training set and 20% as the validation set, with the official test split serving as the testing set. In real-life scenarios, the points in $\boldsymbol{\mathcal{P}}$ have no exact correspondences in $\boldsymbol{\mathcal{Q}}$ [123]. Hence, this experiment uniformly samples 1,024 points from each CAD model twice with different random seeds to generate $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, breaking exact correspondences between the input point clouds, which is distinct from previous works. This experiment follows previous works[11], [15] by randomly applying a rigid transformation along each axis to generate the target points. The rotation along each axis is sampled from $[0, 45°]$ and translation from $[-0.5, 0.5]$. The parameter ν is set to 0.1.

7Scenes [208] is a 3D collection of seven indoor scenes captured using a Kinect RGB-D camera. The dataset comprises 296 scans for training and 57 scans for testing, divided into two parts. Its application is widespread in the evaluation of registration performance using real-world data. To emulate pose variations, the point clouds are uniformly sampled twice and

Table 6.1: Partial-to-Partial Registration results on ModelNet40.

Method	Same-category setup			Cross-category setup		
	MAE(R)	MAE($\boldsymbol{t}$)	CCD	MAE(R)	MAE($\boldsymbol{t}$)	CCD
ICP [42]	10.333	0.1034	0.1066	11.499	0.1084	0.1142
FGR [75]	22.103	0.1273	0.1108	21.928	0.1281	0.1175
RGM [15]	0.8211	0.0094	0.0729	1.3249	0.0164	0.0832
OMNET [123]	2.5944	0.0249	0.0936	3.6001	0.0355	0.1063
RIENet [21]	4.9586	0.0152	0.0735	5.5074	0.0425	0.1072
REGTR [134]	0.7836	**0.0066**	0.0676	0.9105	0.0071	0.0645
CPD [72]	11.033	0.1139	0.1110	12.681	0.1153	0.1154
GMMReg [70]	13.677	0.1344	0.1180	14.899	0.1440	0.1268
SVR [231]	11.857	0.1162	0.1170	13.120	0.1225	0.1237
FilterReg [74]	20.363	0.1558	0.1182	20.531	0.1646	0.1302
DeepGMR [16]	6.8043	0.0683	0.1182	7.3139	0.0718	0.1207
OGMM (ours)	**0.5892**	0.0079	**0.0493**	**0.6309**	**0.0055**	**0.0548**

subjected to a rigid transformation on one of the samples. The rigid transformation is randomly generated by sampling rotation along each axis within the range of $[0, 45°]$ and translation within the range of $[-0.5, 0.5]$. $\nu = 0.5$.

ICL-NUIM [230] is collected from RGB-D scans from the Augmented ICL-NUIM dataset [234]. As proposed in [16], the scenes in ICL-NUIM are divided into 1,278 scans for training and 200 scans for testing. The parameter ν is set to 0.5.

6.3.4 Evaluation on ModelNet40

Same-category setup. This experiment follows the protocol in [134] to generate partial-to-partial point cloud pairs, which are closer to real-world applications. It first generates a half-space with a random direction for each point cloud and shifts it to retain approximately 70% of the points, i.e., 717 points. Tab. 6.1 reports the results. the proposed method significantly outperforms both traditional and deep learning-based methods. Unlike DeepGMR, the proposed method can better detect overlapping regions thanks to the newly introduced overlap scores. Fig. 6.6 shows various successful and unsuccessful registration results. It can be observed that cases with low overlap (e.g., guitar) and with repetitive structures (e.g., net) can be handled by OGMM. The most frequent unsuccessful cases involve point clouds of symmetric objects that can have multiple correct registration solutions along the symmetry axis that do not match with the ground-truth one. This is an intrinsic problem of symmetric cases [123].

Table 6.2: Registration results on ModelNet40 with jittering noise or density variation.

Method	Jittering			Density variation		
	MAE(R)	MAE($\boldsymbol{t}$)	CCD	MAE(R)	MAE($\boldsymbol{t}$)	CCD
ICP [42]	10.609	0.1058	0.1087	5.6654	0.0638	0.0908
FGR [75]	24.534	0.1413	0.1168	16.867	0.0295	0.0890
RGM [15]	1.3536	0.0261	0.0447	1.5948	0.0103	0.0672
OMNET [123]	2.6996	0.0259	0.0977	3.1372	0.0313	0.1010
RIENet [21]	5.8212	0.0635	0.1003	5.5928	0.0725	0.1059
REGTR [134]	1.0984	0.0080	0.0732	3.6409	0.0237	0.0946
CPD [72]	11.049	0.1141	0.1120	8.5349	0.0793	0.0941
GMMReg [70]	13.763	0.1343	0.1187	8.1247	0.0853	0.1024
SVR [231]	12.174	0.1192	0.1178	5.0848	0.0594	0.0949
FilterReg [74]	19.921	0.1548	0.1184	20.098	0.1016	0.1109
DeepGMR [16]	8.8242	0.0699	0.1210	7.1273	0.0689	0.1206
OGMM (ours)	**0.9111**	**0.0071**	**0.0645**	**1.3523**	**0.0100**	**0.0534**

The second most frequent unsuccessful cases involve point cloud pairs with repetitive local geometric structures. These make GMM clustering underperform because features of similar structures in different locations have a small distance in the feature space.

Cross-category setup. This experiment aims to assess the generalization ability of the proposed method by training it and other deep learning-based methods on 20 categories (i.e., airplane, bathtub, bed, bench, bookshelf, bottle, bowl, car, chair, cone, cup, curtain, desk, door, dresser, flower pot, glass box, guitar, keyboard, and lamp), and test them on other different 20 categories (i.e., laptop, mantel, monitor, nightstand, person, piano, plant, radio, range hood, sink, sofa, stairs, stool, table, tent, toilet, TV stand, vase, wardrobe, and Xbox). The data preprocessing remains the same as in the first experiment. Table 6.1 reveals that all learning-based methods perform worse when tested on categories that were not used for training. Nevertheless, the proposed method still outperforms the other baselines in this scenario.

Noisy shapes. To assess the robustness of our approach, Gaussian noise with a mean of 0 and a standard deviation of 0.01 is independently added to each point coordinate and clipped between -0.05 and 0.05. The experiment trains and tests OGMM using the noisy data from ModelNet40. The results from this experiment, shown in the left part of Table 6.2 (Jittering), demonstrate that OGMM can effectively handle noise. This occurs thanks to the proposed cluster-based network that can extract more robust features.

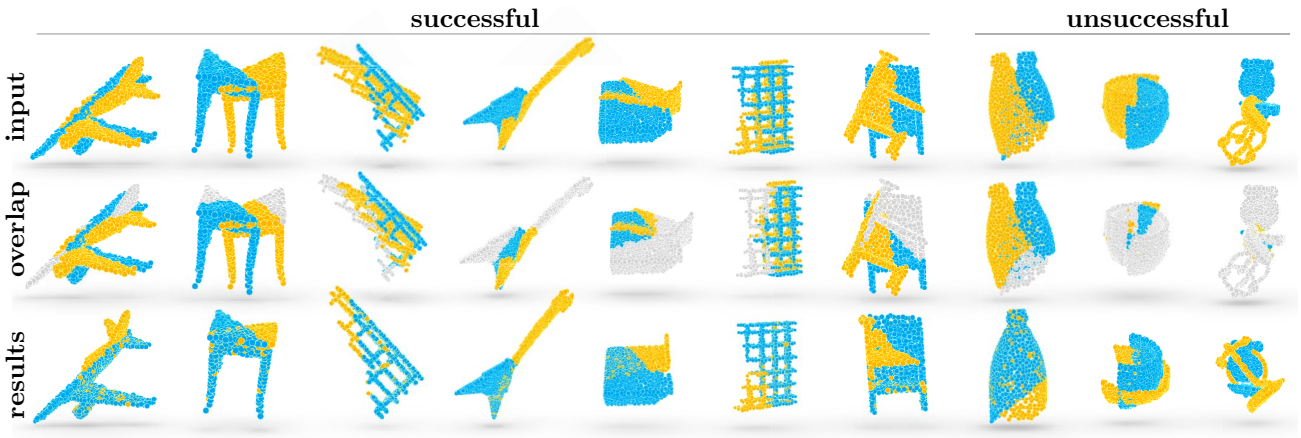


Figure 6.6: Successful and unsuccessful registration results on ModelNet40 using OGMM. The non-overlap regions are shown in grey.

Density variation. The objective of this experiment is to test the resilience of various methods in coping with variations in point cloud densities. In order to do this, points are randomly removed from one point cloud, and the remaining points are replicated to maintain the same number of points, thereby creating a pair of point clouds with differing densities. Table 6.2 (Density variation) showcases the results when the point density is reduced to 50%. It can be seen that the proposed method outperforms the other methods when there are disparities in the densities of the source and target point clouds. This is largely due to the fact that the source and target point clouds are modeled as Gaussian Mixture Models (GMMs), thus making the alignment between the two GMMs less susceptible to density variations than relying on explicit point correspondences.

Complete-to-complete setup. This section first evaluates the complete-to-complete registration performance on ModelNet40 with Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$ and follows the sampling and transformation settings. Tab. 6.3 shows that the proposed method can outperform the GMM-based baselines on this setup. This shows that even though the proposed method is designed for registration, it can also be used with a complete-to-complete setup.

Complete-to-partial setup. This section evaluates the complete-to-partial registration performance on ModelNet40 with Gaussian noise. This section crops the generated source point cloud to create a new source point cloud with approximate overlap ratios of 70%, which includes 717 points. This section randomly draws a rigid transformation along each axis to transform the target point cloud, which contains 1024 points. Tab. 6.3 demonstrates that the proposed approach can achieve better results compared to the GMM-based baselines even in this scenario. The performance improvements over the second-best method, DeepGMR, are significant

Table 6.3: Registration results on ModelNet40 with complete-to-complete and complete-to-partial setups.

Method	Complete-to-complete setup			Complete-to-partial setup		
	MAE(R)	MAE($\boldsymbol{t}$)	CCD	MAE(R)	MAE($\boldsymbol{t}$)	CCD
CPD [72]	0.8171	0.0050	0.0037	10.293	0.0767	0.1118
GMMReg [70]	7.7326	0.0508	0.0837	24.318	0.2578	0.1119
SVR [231]	7.8047	0.0592	0.0744	24.063	0.2480	0.0947
FilterReg [74]	3.4899	0.0247	0.0605	30.653	0.2676	0.1197
DeepGMR [16]	2.2736	0.0150	0.0503	12.612	0.1527	0.1266
OGMM	**0.1461**	**0.0021**	**0.4237**	**7.2820**	**0.0633**	**0.1142**

across all metrics, indicating that integrating overlap scores into the GMM module can enhance registration performance.

6.3.5 Evaluation on 7Scenes and ICL-NUIM

This section conducts experiments on two indoor scenes: real-world 7Scenes [208] and ICL-NUIM. To create source and target counterparts, the original point clouds are uniformly sampled to obtain point clouds with consistent distribution. Following [233], this experiment resamples 2,048 points from each model two times with different random seeds to generate $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$, then create a half-space with a random direction for each point cloud and shift it to retain approximately 70% of the points (1,433) to generate the partial data. As shown in Tab. 6.4, the proposed method achieves the lowest errors on all the metrics on both datasets. OGMM outperforms DeepGMR and GMMReg thanks to the clustered attention network to detect overlapping regions and produce more distinctive features. Fig. 6.7 shows successful and unsuccessful registration results, where the overlap between the two point clouds is 70%. For the unsuccessful case as shown in Fig. 6.8, pairs with repetitive local geometric structures lead to features of similar structures in different locations having a small distance in the feature space.

6.3.6 Ablation Studies

Components of the proposed method. This section analyzes the effectiveness of the proposed method components in the case of partial-to-partial registration (same-category setup). This section assesses the three key novel components of OGMM: Spherical Positional Encoding (SPE), Cluster-based Self-Attention (CSA), and Overlap Score Prediction (OSP). Tab. 6.5 shows that OGMM underperforms when the overlap scores are not used. CSA and SPE mod-

Table 6.4: The registration results on 7Scenes and ICL-NUIM. The best results are bold.

Method	7Scenes			ICL-NUIM		
	MAE(R)	MAE($\boldsymbol{t}$)	CCD	MAE(R)	MAE($\boldsymbol{t}$)	CCD
ICP [42]	18.266	0.2346	0.0793	10.539	0.3301	0.1410
FGR [75]	1.1736	0.0198	0.0270	0.8792	0.0332	0.0835
RGM [15]	3.0334	0.0445	0.0425	1.3279	0.0416	0.0840
OMNET [123]	9.8499	0.1416	0.1879	17.177	0.4587	0.1947
REGTR [134]	4.6143	0.0827	0.1733	3.2503	0.1044	0.0902
CPD [72]	4.9897	0.1056	0.0900	9.8322	0.3828	0.1581
GMMReg [70]	11.081	0.1213	0.1374	6.5411	0.1700	0.1530
SVR [231]	10.729	0.1152	0.1388	6.3229	0.1946	0.1528
FilterReg [74]	18.113	0.2521	0.0636	28.317	0.7930	0.1693
DeepGMR [16]	8.8478	0.1534	0.0541	6.4600	0.1899	0.1269
OGMM (ours)	**0.5764**	**0.0088**	**0.0214**	**0.6279**	**0.0305**	**0.0732**

Table 6.5: Ablation study on ModelNet40.

SPE	CSA	OSP	MAE(R)	MAE($\boldsymbol{t}$)	CCD
	✓	✓	0.9534	0.0096	0.0589
✓		✓	1.9060	0.0169	0.0889
✓	✓		6.7087	0.0729	0.1155
✓	✓	✓	**0.5892**	**0.0079**	**0.0493**

ules help achieve a higher registration accuracy, allowing for more distinctive features to be produced.

Loss functions. This section trains the proposed model with different combinations of the Global Registration loss (GR), the Clustering-based loss (GS), and the Overlap Score loss (OS). Experiments are conducted on ModelNet40 (same-category setup). Tab. 6.6 shows that the combination of GR and OS losses provides the major contribution.

Inference time. This experiment evaluates the efficiency of OGMM and compares it to other approaches on ModelNet40 (same-category setup). This experiment averages the inference time of the proposed method using a single Tesla V100 GPU (32G) and two Intel(R) 6226 CPUs. f and c represent the full and cluster-based attention. Tab. 6.7 reports the results. Compared to RGM, OGMM utilizes similar network architecture but different matching strategies and

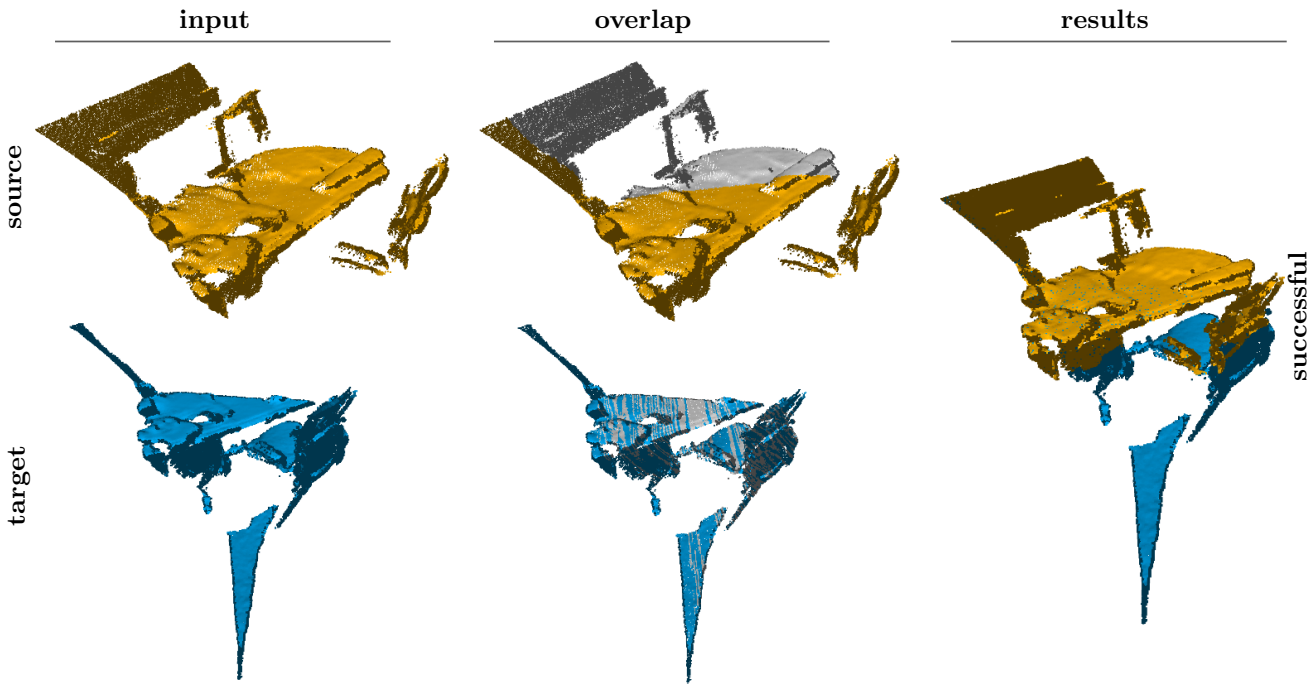


Figure 6.7: Qualitative successful results on the 7scenes dataset.

Table 6.6: Loss function analysis on ModelNet40.

GR	CL	OS	MAE(R)	MAE($\boldsymbol{t}$)	CCD
✓			3.7284	0.0272	0.0958
	✓		7.0198	0.0553	0.1133
		✓	4.4373	0.0392	0.1006
✓	✓		3.2335	0.0262	0.0939
	✓	✓	2.6470	0.0250	0.0751
✓		✓	0.7828	0.0085	0.0515
✓	✓	✓	**0.5892**	**0.0079**	**0.0493**

attention modules. OGMM (c) outperforms RGM and reduces about 9× the computation time. Compared with full attention, cluster-based attention can speed up 8× times, which verifies the effectiveness of the devised clustered strategy in reducing the computing complexity. The proposed method is inferior to DeepGRM because the overlap detection module is introduced to handle partial overlap cases.

Different overlapping ratios. Because the overlap ratio may affect registration performance, this section analyzes the performance variation when the overlap ratio decreases gradually. This experiment evaluates the performance of the proposed method on noisy ModelNet40. It utilizes the same crop setting as RPMNet to generate point clouds with approximate overlap ratios of 70%, 60%, 50%, 40%, and 30%, respectively. The model is trained on data with a

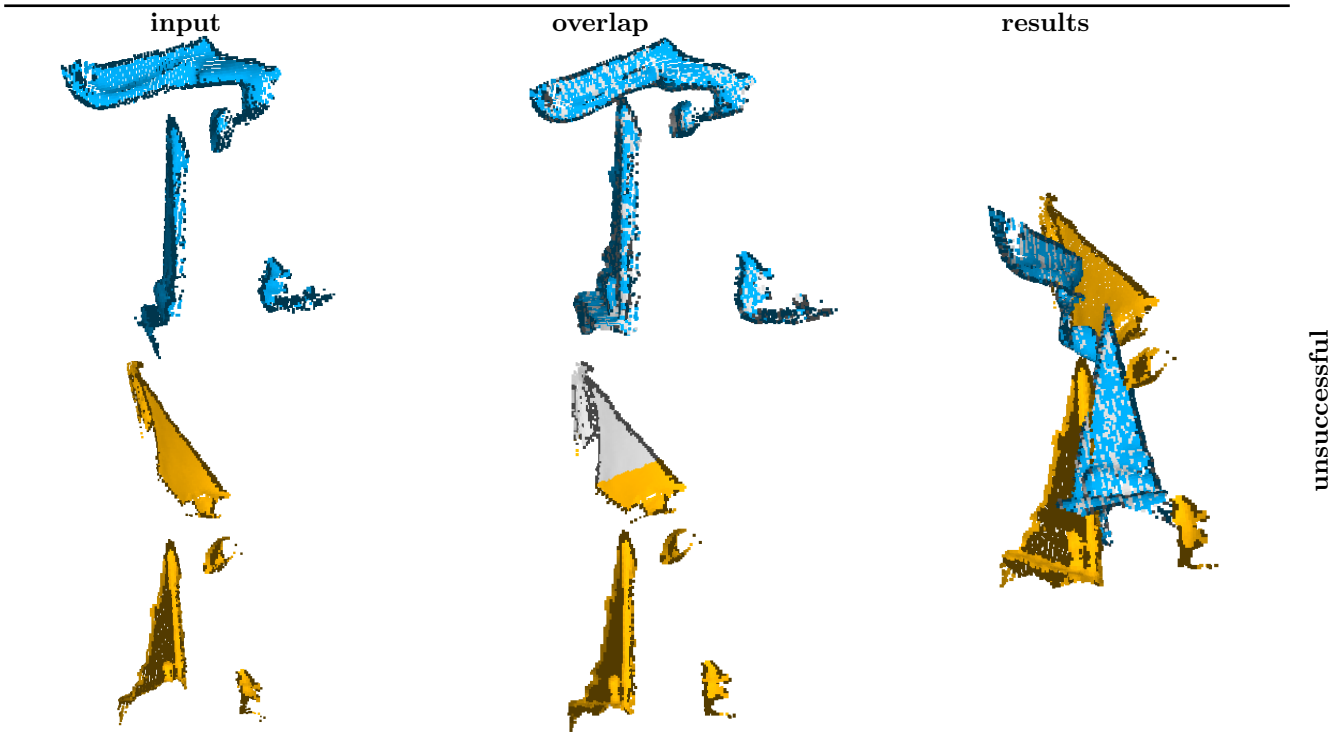


Figure 6.8: Qualitative unsuccessful results on the 7scenes dataset.

Table 6.7: Comparisons of the average inference time.

Method	CPD	GMMReg	RGM	DeepGMR	SoftClu (f)	SoftClu (c)
Time(s)	4.347	2.536	0.057	**0.002**	0.047	0.006

50% overlap ratio and tested on different overlap ratios. Tab. 6.8 shows the registration results under different overlap ratios with Gaussian noise sampled from $\mathcal{N}(0, 0.01)$ and clipped to $[-0.05, 0.05]$. The lower the overlap ratio, the higher the registration error.

Table 6.8: The effects of the overlap ratio on ModelNet40.

Ratio	30%	40%	50%	60%	70%
MAE (R)	5.6462	3.5995	2.1000	1.2067	0.9111
MAE ($\boldsymbol{t}$)	0.1220	0.0716	0.0178	0.0159	0.0071
CCD	0.1097	0.0957	0.0937	0.0664	0.0645

Different cluster numbers. Table 6.9 illustrates the performance of the proposed method under various cluster numbers, including 8, 16, 32, 48, and 64, with a 50% overlap ratio. The results indicate that a lower registration error is achieved with a larger number of clusters, ranging from 32 to 64. When the number of clusters is small (e.g. 8, 16), a larger registration error is observed.

Table 6.9: The effects of the cluster numbers on ModelNet40 with 50% overlapping ratio and Gaussian noise.

Ratio	8	16	32	48	64
MAE (R)	5.6462	3.5995	2.1625	2.0834	2.1000
MAE ($\boldsymbol{t}$)	0.1220	0.0716	0.0187	0.0181	0.0178
CCD	0.1097	0.0957	0.0942	0.0885	0.0937

6.4 Summary and conclusions

This chapter introduces a learning-based probabilistic registration method for point clouds with partial overlaps. It utilizes a clustered attention-based network to identify the overlap regions between the two point clouds and formulates the registration process as the reduction of the discrepancy between Gaussian mixtures, guided by the overlap information. The results of the experiments demonstrate that the proposed method outperforms both traditional and deep learning-based registration techniques in various data scenarios. Additionally, the OGMM is robust to noise and can effectively generalize to different objects and real-world data. This method provides a novel integration of 3D neural networks within a probabilistic registration framework. Future work will focus on developing an unsupervised probabilistic registration method for detecting overlap regions to reduce the reliance on labeled data.

Chapter 7

Conclusions and Future Work

7.1 Summary of Contribution and Outcomes

This thesis focused on developing optimization and learning-based point cloud registration methods to handle point cloud pairs with large rotations, partial overlaps, and density variation. This thesis also tried to provide unsupervised learning approaches to reduce networks' dependence on annotations. The main contribution and outcomes of my Ph.D. work are summarized as follows:

Chapter 3 devised an augmentation-free unsupervised approach for point clouds to learn transferable point-level features via soft clustering, named SoftClu. SoftClu assumes that the points belonging to a cluster should be close to each other in both geometric and feature spaces. This differs from typical contrastive learning, which builds similar representations for a whole point cloud and its augmented versions. SoftClu exploits the affiliation of points to their clusters as a proxy to enable self-training through a pseudo-label prediction task. Under the constraint that these pseudo-labels induce the equipartition of the point cloud, SoftClu is casted as an optimal transport problem, which can be solved by using an efficient variant of the Sinkhorn-Knopp algorithm. SoftClu formulates an unsupervised loss to minimize the standard cross-entropy between pseudo-labels and predicted labels. In addition to its core contributions, SoftClu also opens up possibilities for extending correspondence-free methods to tackle partial overlapping registration, as demonstrated in Chapter 4.

Chapter 4 provided an unsupervised correspondence-free method to solve point cloud registration with large rotations and partial overlaps. The proposed approach combines unsupervised feature learning with a beam search scheme in the 3D rotation space, which can adjust well to the case of large rotation. To handle point clouds with partial overlaps, a modified version of SoftClu is applied to segment both the source and target point clouds into discrete geometric

partitions. Subsequently, the registration process involves iterative use of the IC-LK algorithm to minimize the distance between the feature descriptors of corresponding partitions. However, using only global features might not sufficiently capture intricate local variations and spatial relationships within the point cloud, especially when dealing with complex structures or low partial overlaps. As a result, this may lead to suboptimal or inaccurate registration outcomes.

Chapter 5 proposed a learning framework by simultaneously considering point-wise and structural matchings to estimate correspondences in a coarse-to-fine manner. This approach aims to address scenarios where correspondence-free methods fail due to point clouds with complex structures or low partial overlaps. The method transforms these two types of matchings into optimizations based on Wasserstein distance and Gromov-Wasserstein distance, respectively, effectively reshaping the task of establishing correspondences into a fused optimal transport problem. Additionally, an overlap attention module, primarily composed of transformer layers, is introduced to predict the likelihood (overlap score) of each point belonging to the overlapping region. This overlap score then guides the correspondence prediction process, enhancing the accuracy of the registration procedure. Nevertheless, the method encounters challenges when dealing with point clouds exhibiting variant densities.

Chapter 6 proposed a novel overlap-guided probabilistic registration approach, which addressed the issue of variant densities in point clouds. This approach reframed the registration problem as aligning two Gaussian mixtures (GMM) to minimize statistical differences between corresponding mixtures. It utilized an overlap Transformer module based on clustering to embed cross-point cloud information, facilitating the detection of overlap regions under the guidance of overlap scores. To reduce computation complexity, a cluster-based loss was introduced, ensuring the network learned a consistent Gaussian Mixture Model representation across feature and geometric spaces rather than fitting a GMM in a single feature space.

From Chapter 3 to Chapter 6, each chapter of this thesis is backed by a minimum of one published paper[1] listed in the List of Publications. Therefore, all works proposed in this thesis are significant in to point cloud registration field.

7.2 Recommendations & Future Work

This section recommends some future research opportunities related to point cloud registration.

- Many registration works show that extracting distinctive local features using pure geometric knowledge is difficult. It would be beneficial to fuse 2D features into learned 3D geometric features or transfer information from 2D domains into 3D domains.

[1]The chapters 3, 4, and 6 of the thesis have been published, while one paper from chapter 5 is currently under revision (TPAMI's comments are revise and resubmitted as new)

- Correspondence-based methods are still dominant in current registration advances; thus, designing more sophisticated algorithms to estimate correspondence is also a practicable alternative to improve registration accuracy. Besides, predicting correspondences requires that the learned features are invariant to transformation. Therefore, devising algorithms to extract robust and rotation-invariant is a promising research direction.
- With the development of 3D devices, more and more applications require the integration of the point clouds captured from devices, which is involved in registration across different sources with density variations. Therefore, developing robust matching methods to handle cross-source registration tasks is significant.
- The registration of point clouds of large-scale outdoor scenes is a challenging research problem due to the complexity of the environment, the presence of occlusions, and the limitations of available sensors. Exploring methods for handling these challenges and creating a more accurate representation of outdoor scenes is beneficial.
- Point clouds can be more useful if they are semantically enriched, i.e., if they contain information about the objects and scenes they represent. Researchers can investigate how to incorporate semantic information into registration algorithms to improve their accuracy and efficiency.

Bibliography

[1] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "Pointnetlk: Robust & efficient point cloud registration using pointnet," in *CVPR*, 2019, pp. 7163–7172.

[2] Y. Wang and J. M. Solomon, "Prnet: Self-supervised learning for partial-to-partial registration," in *NeurIPS*, 2019, pp. 8812–8824.

[3] Z. Zhang, Y. Dai, and J. Sun, "Deep learning based point cloud registration: An overview," *VR&IH*, vol. 2, no. 3, pp. 222–246, 2020.

[4] W. Liu, H. Wu, and G. S. Chirikjian, "Lsg-cpd: Coherent point drift with local surface geometry for point cloud registration," in *ICCV*, 2021, pp. 15 293–15 302.

[5] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, 1996.

[6] P. J. Bes, N. D. McKay, *et al.*, "A method for registration of 3-d shapes," *TPAMI*, vol. 14, no. 2, pp. 239–256, 1992.

[7] J. J. Moré, "The levenberg-marquardt algorithm: Implementation and theory," in *Numerical Anal.* Springer, 1978, pp. 105–116.

[8] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp.," in *RSS*, Seattle, WA, vol. 2, 2009, p. 435.

[9] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *CVPR*, 2022, pp. 11 143–11 152.

[10] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *ICCV*, 2019, pp. 3523–3532.

[11] X. Huang, G. Mei, and J. Zhang, "Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences," in *CVPR*, 2020.

[12] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *CVPR*, 2020, pp. 2514–2523.

[13] X. Bai, Z. Luo, L. Zhou, *et al.*, "Pointdsc: Robust point cloud registration using deep spatial consistency," in *CVPR*, 2021.

[14] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *CVPR*, 2021.

[15] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *CVPR*, 2021.

[16] W. Yuan, B. Eckart, K. Kim, V. Jampani, D. Fox, and J. Kautz, "Deepgmr: Learning latent gaussian mixture models for registration," in *ECCV*, Springer, 2020, pp. 733–750.

[17] G. Mei, X. Huang, J. Zhang, and Q. Wu, "Partial point cloud registration via soft segmentation," in *ICIP*, IEEE, 2022, pp. 681–685.

[18] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *ICCV*, 2019, pp. 8958–8966.

[19] A. Zeng, S. Song, M. Nießner, *et al.*, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.

[20] X. Li, J. Sun, C.-M. Own, and W. Tao, "Gaussian mixture model-based registration network for point clouds with partial overlap," in *ICANN*, Springer, 2022, pp. 405–416.

[21] Y. Shen, L. Hui, H. Jiang, J. Xie, and J. Yang, "Reliable inlier evaluation for unsupervised point cloud registration," in *AAAI*, 2022.

[22] X. Huang, G. Mei, J. Zhang, and R. Abbas, "A comprehensive survey on point cloud registration," *arXiv preprint arXiv:2103.02690*, 2021.

[23] G. Mei, X. Huang, J. Zhang, and Q. Wu, "Overlap-guided coarse-to-fine correspondence prediction for point cloud registration," in *ICME*, IEEE, 2022, pp. 1–6.

[24] X. Huang, G. Mei, and J. Zhang, "Cross-source point cloud registration: Challenges, progress and prospects," *Neurocomputing*, p. 126 383, 2023.

[25] G. Mei, H. Tang, X. Huang, *et al.*, "Unsupervised deep probabilistic approach for partial point cloud registration," in *CVPR*, 2023, pp. 13 611–13 620.

[26] H. Deng, T. Birdal, *et al.*, "Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors," in *ECCV*, 2018, pp. 602–618.

[27] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany, "Pointcontrast: Unsupervised pre-training for 3d point cloud understanding," in *ECCV*, Springer, 2020, pp. 574–591.

[28] M. El Banani, L. Gao, and J. Johnson, "Unsupervisedr&r: Unsupervised point cloud registration via differentiable rendering," in *CVPR*, 2021, pp. 7129–7139.

[29] M. Li, Y. Xie, Y. Shen, *et al.*, "Hybridcr: Weakly-supervised 3d point cloud semantic segmentation via hybrid contrastive regularization," in *CVPR*, 2022, pp. 14 930–14 939.

[30] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *CVPR*, 2021.

[31] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017, pp. 1802–1811.

[32] G. Mei, F. Poiesi, C. Saltori, J. Zhang, E. Ricci, and N. Sebe, "Overlap-guided gaussian mixture models for point cloud registration," in *WACV*, 2023, pp. 4511–4520.

[33] F. Pernkopf and D. Bouchaffra, "Genetic-based em algorithm for learning gaussian mixture models," *TPAMI*, vol. 27, no. 8, pp. 1344–1348, 2005.

[34] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[35] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *CGF*, Wiley Online Library, vol. 26, 2007, pp. 214–226.

[36] G. Mei, "Point cloud registration with self-supervised feature learning and beam search," in *DICTA*, 2021, pp. 01–08.

[37] X. Huang, J. Zhang, Q. Wu, L. Fan, and C. Yuan, "A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds," *TCSVT*, vol. 28, no. 10, pp. 2965–2977, 2017.

[38] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *ICRA*, 2012, pp. 1697–1702.

[39] Y. Zhang, X. Xiong, M. Zheng, and X. Huang, "Lidar strip adjustment using multifeatures matched with aerial images," *TGROS*, vol. 53, no. 2, pp. 976–987, 2014.

[40] Y. Park, V. Lepetit, and W. Woo, "Multiple 3d object tracking for augmented reality," in *ISMAR*, 2008, pp. 117–120.

[41] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, 1981.

[42] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, SPIE, vol. 1611, 1992, pp. 586–606.

[43] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *CGF*, Wiley Online Library, vol. 32, 2013, pp. 113–123.

[44] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," *IVC*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[45] F. D. Foresee and M. T. Hagan, "Gauss-newton approximation to bayesian learning," in *ICNN*, IEEE, vol. 3, 1997, pp. 1930–1935.

[46] J. Yang, H. Li, and Y. Jia, "Go-icp: Solving 3d registration efficiently and globally optimally," in *ICCV*, 2013, pp. 1457–1464.

[47] L. Mitten, "Branch-and-bound methods: General formulation and properties," *OR*, vol. 18, no. 1, pp. 24–34, 1970.

[48] S. Fantoni, U. Castellani, and A. Fusiello, "Accurate and automatic alignment of range surfaces," in *IC3DIM/TPVT*, IEEE, 2012, pp. 73–80.

[49] P. J. Huber, "Robust estimation of a location parameter," in *BIS: M&D*, Springer, 1992, pp. 492–518.

[50] D. Aiger, N. J. Mitra, and D. Cohen-Or, "4-points congruent sets for robust pairwise surface registration," in *ACM SIGGRAPH*, 2008, pp. 1–10.

[51] P. H. Torr and A. Zisserman, "Mlesac: A new robust estimator with application to estimating image geometry," *CVIU*, vol. 78, no. 1, pp. 138–156, 2000.

[52] D. Barath, J. Noskova, M. Ivashechkin, and J. Matas, "Magsac++, a fast, reliable and accurate robust estimator," in *CVPR*, 2020, pp. 1304–1312.

[53] K. Ni, H. Jin, and F. Dellaert, "Groupsac: Efficient consensus in the presence of groupings," in *ICCV*, IEEE, 2009, pp. 2193–2200.

[54] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Joint Pattern Recognition Symposium*, Springer, 2003, pp. 236–243.

[55] D. Barath and J. Matas, "Graph-cut ransac," in *CVPR*, 2018, pp. 6733–6741.

[56] J. Li, Q. Hu, and M. Ai, "Point cloud registration based on one-point ransac and scale-annealing biweight estimation," *TGRS*, vol. 59, no. 11, pp. 9716–9729, 2021.

[57] E. Brachmann, A. Krull, S. Nowozin, *et al.*, "Dsac-differentiable ransac for camera localization," in *CVPR*, 2017, pp. 6684–6692.

[58] E. Brachmann and C. Rother, "Neural-guided ransac: Learning where to sample model hypotheses," in *ICCV*, 2019, pp. 4322–4331.

[59] Á. P. Bustos and T.-J. Chin, "Guaranteed outlier removal for point cloud registration with correspondences," *TPAMI*, vol. 40, no. 12, pp. 2868–2882, 2017.

[60] L. Livi and A. Rizzi, "The graph matching problem," *Pattern Anal. Appl.*, vol. 16, no. 3, pp. 253–283, 2013.

[61] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce, "A tensor-based algorithm for high-order graph matching," *TPAMI*, vol. 33, no. 12, pp. 2383–2395, 2011.

[62] E. M. Loiola, N. M. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido, "A survey for the quadratic assignment problem," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 657–690, 2007.

[63] M. Leordeanu and M. Hebert, "A spectral technique for correspondence problems using pairwise constraints," in *ICCV*, IEEE, vol. 2, 2005, pp. 1482–1489.

[64] H. Almohamad and S. O. Duffuaa, "A linear programming approach for the weighted graph matching problem," *TPAMI*, vol. 15, no. 5, pp. 522–525, 1993.

[65] F. Zhou and F. De la Torre, "Factorized graph matching," in *CVPR*, IEEE, 2012, pp. 127–134.

[66] R. Zass and A. Shashua, "Probabilistic graph and hypergraph matching," in *CVPR*, IEEE, 2008, pp. 1–8.

[67] H. Zhu, C. Cui, L. Deng, R. C. Cheung, and H. Yan, "Elastic net constraint-based tensor model for high-order graph matching," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 4062–4074, 2019.

[68] X. Huang, J. Zhang, L. Fan, Q. Wu, and C. Yuan, "A systematic approach for cross-source point cloud registration by preserving macro and micro structures," *TIP*, vol. 26, no. 7, pp. 3261–3276, 2017.

[69] B. Eckart, K. Kim, and J. Kautz, "Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration," in *ECCV*, 2018, pp. 705–721.

[70] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *TPAMI*, vol. 33, no. 8, pp. 1633–1645, 2010.

[71] F. J. Lawin, M. Danelljan, F. S. Khan, P.-E. Forssén, and M. Felsberg, "Density adaptive point set registration," in *CVPR*, 2018, pp. 3829–3837.

[72] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *TPAMI*, vol. 32, no. 12, pp. 2262–2275, 2010.

[73] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *TPAMI*, vol. 40, no. 6, pp. 1397–1410, 2017.

[74] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *CVPR*, 2019, pp. 11 095–11 104.

[75] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *ECCV*, Springer, 2016, pp. 766–782.

[76] J. T. Barron, "A general and adaptive robust loss function," in *CVPR*, 2019, pp. 4331–4339.

[77] H. Yang, J. Shi, and L. Carlone, "Teaser: Fast and certifiable point cloud registration," *T-RO*, 2020.

[78] W. Chen, H. Li, Q. Nie, and Y.-H. Liu, "Deterministic point cloud registration via novel transformation decomposition," in *CVPR*, 2022, pp. 6348–6356.

[79] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017, pp. 652–660.

[80] M. Zaheer, S. Kottur, *et al.*, "Deep sets," in *NeurIPS*, 2017, pp. 3391–3401.

[81] Y. Rao, J. Lu, and J. Zhou, "Global-local bidirectional reasoning for unsupervised representation learning of 3d point clouds," in *CVPR*, 2020.

[82] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NeurIPS*, 2017, pp. 5099–5108.

[83] G. Qian, Y. Li, H. Peng, *et al.*, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *NeurIPS*, vol. 35, pp. 23 192–23 204, 2022.

[84] Y. Li, R. Bu, *et al.*, "Pointcnn: Convolution on x-transformed points," in *NeurIPS*, 2018, pp. 820–830.

[85] W. Wu, Z. Qi, and L. Fuxin, "Pointconv: Deep convolutional networks on 3d point clouds," in *CVPR*, 2019, pp. 9621–9630.

[86] Y. Liu, B. Fan, *et al.*, "Relation-shape convolutional neural network for point cloud analysis," in *CVPR*, 2019, pp. 8895–8904.

[87] Z. Han, X. Wang, *et al.*, "Multi-angle point cloud-vae: Unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction," in *ICCV*, 2019, pp. 10 441–10 450.

[88] Z. Yan, R. Hu, X. Yan, *et al.*, "Rpm-net: Recurrent prediction of motion and parts from point cloud," in *SIGGRAPH Asia*, 2020.

[89] B. Wu, Y. Liu, B. Lang, and L. Huang, "Dgcnn: Disordered graph convolutional neural network based on the gaussian mixture model," *Neurocomputing*, vol. 321, pp. 346–356, 2018.

[90] M. Xu, R. Ding, H. Zhao, and X. Qi, "Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *CVPR*, 2021.

[91] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *CVPR*, 2021.

[92] L. Yu, X. Li, C.-W. Fu, D. Cohen-Or, and P.-A. Heng, "Ec-net: An edge-aware point set consolidation network," in *ECCV*, 2018, pp. 386–402.

[93] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *ICCV*, 2019, pp. 6411–6420.

[94] T. Sun, G. Liu, R. Li, S. Liu, S. Zhu, and B. Zeng, "Quadratic terms based point-to-surface 3d representation for deep learning of point cloud," *TCSVT*, 2021.

[95] D. Ding, C. Qiu, F. Liu, and Z. Pan, "Point cloud upsampling via perturbation learning," *TCSVT*, vol. 31, no. 12, pp. 4661–4672, 2021.

[96] F. Song, Y. Shao, W. Gao, H. Wang, and T. Li, "Layer-wise geometry aggregation framework for lossless lidar point cloud compression," *TCSVT*, 2021.

[97] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[98] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *CVPR*, 2021, pp. 16 259–16 268.

[99] C. Zhang, H. Wan, X. Shen, and Z. Wu, "Patchformer: An efficient point transformer with patch attention," in *CVPR*, 2022, pp. 11 799–11 808.

[100] J. Yang, Q. Zhang, B. Ni, *et al.*, "Modeling point clouds with self-attention and gumbel subset sampling," in *CVPR*, 2019, pp. 3323–3332.

[101] L. Zhao, J. Guo, D. Xu, and L. Sheng, "Transformer3d-det: Improving 3d object detection by vote refinement," *TCSVT*, 2021.

[102] D. Lu, Q. Xie, K. Gao, L. Xu, and J. Li, "3dctn: 3d convolution-transformer network for point cloud classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 24 854–24 865, 2022.

[103] H. Huang and Y. Fang, "Adaptive wavelet transformer network for 3d shape representation learning," in *ICLR*, 2021.

[104] H. Wang, Q. Liu, X. Yue, J. Lasenby, and M. J. Kusner, "Unsupervised point cloud pre-training via view-point occlusion, completion," in *ICCV*, 2020.

[105] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "Unsupervised cycle-consistent deformation for shape matching," in *CGF*, Wiley Online Library, vol. 38, 2019, pp. 123–133.

[106] L. Yang, W. Liu, Z. Cui, N. Chen, and W. Wang, "Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes," in *ECCV*, Springer, 2020, pp. 455–472.

[107] H. Jiang, Y. Shen, J. Xie, J. Li, J. Qian, and J. Yang, "Sampling network guided cross-entropy method for unsupervised point cloud registration," in *ICCV*, 2021, pp. 6128–6137.

[108] S. Huang, Y. Xie, S.-C. Zhu, and Y. Zhu, "Spatio-temporal self-supervised representation learning for 3d point clouds," in *ICCV*, 2021.

[109] B. Eckart, W. Yuan, C. Liu, and J. Kautz, "Self-supervised learning on 3d point clouds by learning discrete generative models," in *CVPR*, 2021.

[110] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018, pp. 206–215.

[111] X. Liu, Z. Han, X. Wen, Y.-S. Liu, and M. Zwicker, "L2g auto-encoder: Understanding point clouds by local-to-global reconstruction with hierarchical self-attention," in *ACM MM*, 2019, pp. 989–997.

[112] S. Chen, C. Duan, Y. Yang, D. Li, C. Feng, and D. Tian, "Deep unsupervised learning of 3d point clouds via graph topology inference and filtering," *TIP*, vol. 29, pp. 3183–3198, 2019.

[113] P. Achlioptas, O. Diamanti, *et al.*, "Learning representations and generative models for 3d point clouds," in *ICML*, 2018, pp. 40–49.

[114] X. Gao, W. Hu, and G.-J. Qi, "Graphter: Unsupervised learning of graph transformation equivariant representations via auto-encoding node-wise transformations," in *CVPR*, 2020.

[115] A. Sanghi, "Info3d: Representation learning on 3d objects using mutual information maximization and contrastive learning," in *ECCV*, Springer, 2020, pp. 626–642.

[116] J. Sauder and B. Sievers, "Self-supervised deep learning on point clouds by reconstructing space," in *NeurIPS*, 2019, pp. 12 942–12 952.

[117] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[118] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *CVPR*, 2022, pp. 19 313–19 322.

[119] H. Deng, T. Birdal, and S. Ilic, "Ppfnet: Global context aware local features for robust 3d point matching," in *CVPR*, 2018, pp. 195–205.

[120] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *ECCV*, 2018, pp. 607–623.

[121] J. Zhou, M. Wang, W. Mao, M. Gong, and X. Liu, "Siamesepointnet: A siamese point network architecture for learning 3d shape descriptor," in *CGF*, Wiley Online Library, vol. 39, 2020, pp. 309–321.

[122] V. Sarode, X. Li, H. Goforth, *et al.*, "Pcrnet: Point cloud registration network using pointnet encoding," *arXiv preprint arXiv:1908.07906*, 2019.

[123] H. Xu, S. Liu, G. Wang, G. Liu, and B. Zeng, "Omnet: Learning overlapping mask for partial-to-partial point cloud registration," in *CVPR*, 2021, pp. 3132–3141.

[124] M. Zhu, M. Ghaffari, and H. Peng, "Correspondence-free point cloud registration with so (3)-equivariant implicit shape representations," in *CoRL*, PMLR, 2022, pp. 1412–1422.

[125] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "A representation separation perspective to correspondence-free unsupervised 3-d point cloud registration," *IEEE GRSL*, vol. 19, pp. 1–5, 2021.

[126] H. Xu, N. Ye, G. Liu, B. Zeng, and S. Liu, "Finet: Dual branches feature interaction for partial-to-partial point cloud registration," in *AAAI*, vol. 36, 2022, pp. 2848–2856.

[127] Z. Gojcic, C. Zhou, J. D. Wegner, and A. Wieser, "The perfect match: 3d point cloud matching with smoothed densities," in *CVPR*, 2019, pp. 5545–5554.

[128] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *CVPR*, Ieee, 2010, pp. 998–1005.

[129] H. Deng *et al.*, "3d local features for direct pairwise registration," in *CVPR*, 2019, pp. 3244–3253.

[130] H. Wang, Y. Liu, Z. Dong, and W. Wang, "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors," in *ACM MM*, 2022, pp. 1630–1641.

[131] J. Yang, C. Zhao, K. Xian, A. Zhu, and Z. Cao, "Learning to fuse local geometric features for 3d rigid data matching," *IF*, vol. 61, pp. 24–35, 2020.

[132] S. A. Ali, K. Kahraman, G. Reis, and D. Stricker, "Rpsrnet: End-to-end trainable rigid point set registration network using barnes-hut 2d-tree representation," in *CVPR*, 2021, pp. 13 100–13 110.

[133] Z. J. Yew *et al.*, "Rpm-net: Robust point matching using learned features," in *CVPR*, 2020, pp. 11 824–11 833.

[134] Z. J. Yew and G. H. Lee, "Regtr: End-to-end point cloud correspondences with transformers," in *CVPR*, 2022, pp. 6677–6686.

[135] Z. Su, Y. Wang, R. Shi, *et al.*, "Optimal mass transport for shape matching and comparison," *TPAMI*, vol. 37, no. 11, pp. 2246–2259, 2015.

[136] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport: With applications to data science," *FTML*, vol. 11, no. 5-6, pp. 355–607, 2019.

[137] B. Wu, J. Ma, G. Chen, and P. An, "Feature interactive representation for point cloud registration," in *ICCV*, 2021, pp. 5530–5539.

[138] Z. Zhang, J. Sun, Y. Dai, B. Fan, and M. He, "Vrnet: Learning the rectified virtual corresponding points for 3d point cloud registration," *TCSVT*, vol. 32, no. 8, pp. 4997–5010, 2022.

[139] G. Chen, M. Wang, Q. Zhang, L. Yuan, and Y. Yue, "Ftcsvt," *TNNLS*, 2023.

[140] Z. Zhang, J. Sun, Y. Dai, D. Zhou, X. Song, and M. He, "End-to-end learning the partial permutation matrix for robust 3d point cloud registration," in *AAAI*, vol. 36, 2022, pp. 3399–3407.

[141] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepicp: An end-to-end deep neural network for 3d point cloud registration," in *ICCV*, 2019.

[142] J. Li, C. Zhang, Z. Xu, H. Zhou, and C. Zhang, "Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration," in *ECCV*, 2020.

[143] K. Fischer, M. Simon, F. Olsner, S. Milz, H.-M. Gross, and P. Mader, "Stickypillars: Robust and efficient feature matching on point clouds using graph neural networks," in *CVPR*, 2021, pp. 313–323.

[144] F. Lu, G. Chen, Y. Liu, *et al.*, "Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration," in *ICCV*, 2021, pp. 16 014–16 023.

[145] Z. Dang, F. Wang, and M. Salzmann, "Learning 3d-3d correspondences for one-shot partial-to-partial registration," *arXiv preprint arXiv:2006.04523*, 2020.

[146] Z. Shen and et al., "Accurate point cloud registration with robust optimal transport," in *NeurIPS*, 2021.

[147] N. Lang and J. M. Francos, "Deepume: Learning the universal manifold embedding for robust point cloud registration," *arXiv preprint arXiv:2112.09938*, 2021.

[148] Z. Chen, H. Chen, L. Gong, *et al.*, "Utopic: Uncertainty-aware overlap prediction network for partial point cloud registration," in *CGF*, Wiley Online Library, vol. 41, 2022, pp. 87–98.

[149] A.-Q. Cao, G. Puy, A. Boulch, and R. Marlet, "Pcam: Product of cross-attention matrices for rigid registration of point clouds," in *ICCV*, 2021, pp. 13 229–13 238.

[150] H. Yu and et al., "Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration," *NeurIPS*, vol. 34, 2021.

[151] Z. Chen, F. Yang, and W. Tao, "Detarnet: Decoupling translation and rotation by siamese network for point cloud registration," in *AAAI*, vol. 36, 2022, pp. 401–409.

[152] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo, "3dregnet: A deep neural network for 3d point registration," in *CVPR*, 2020, pp. 7193–7203.

[153] Z. Chen, K. Sun, F. Yang, and W. Tao, "Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration," in *CVPR*, 2022, pp. 13 221–13 231.

[154] L. Sun and L. Deng, "Trivoc: Efficient voting-based consensus maximization for robust point cloud registration with extreme outlier ratios," *RA-L*, vol. 7, no. 2, pp. 4654–4661, 2022.

[155] J. Lee, S. Kim, M. Cho, and J. Park, "Deep hough voting for robust global registration," in *ICCV*, 2021, pp. 15 994–16 003.

[156] M. Yuan, Z. Li, Q. Jin, X. Chen, and M. Wang, "Pointclm: A contrastive learning-based framework for multi-instance point cloud registration," in *ECCV*, Springer, 2022, pp. 595–611.

[157] W. Tang and D. Zou, "Multi-instance point cloud registration by efficient correspondence clustering," in *CVPR*, 2022, pp. 6667–6676.

[158] L. Chapel, M. Z. Alaya, and G. Gasso, "Partial optimal tranport with applications on positive-unlabeled learning," *NeurIPS*, vol. 33, pp. 2903–2913, 2020.

[159] J. Solomon, F. De Goes, G. Peyré, *et al.*, "Convolutional wasserstein distances: Efficient optimal transportation on geometric domains," *ACM TOG*, vol. 34, no. 4, pp. 1–11, 2015.

[160] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty, "Fused gromov-wasserstein distance for structured objects," *Algorithms*, vol. 13, no. 9, p. 212, 2020.

[161] J. Solomon, G. Peyré, V. G. Kim, and S. Sra, "Entropic metric alignment for correspondence problems," *ACM TOG*, vol. 35, no. 4, pp. 1–13, 2016.

[162] H. Xu, D. Luo, H. Zha, and L. C. Duke, "Gromov-wasserstein learning for graph matching and node embedding," in *ICML*, PMLR, 2019, pp. 6932–6941.

[163] V. Titouan, N. Courty, R. Tavenard, and R. Flamary, "Optimal transport for structured data with application on graphs," in *ICML*, PMLR, 2019, pp. 6275–6284.

[164] H. Liu, X. Gu, and D. Samaras, "Wasserstein gan with quadratic transport cost," in *ICCV*, 2019, pp. 4832–4841.

[165] N. Courty, R. Flamary, and D. Tuia, "Domain adaptation with regularized optimal transport," in *ECML-PKDD*, Springer, 2014, pp. 274–289.

[166] M. Tang, C. Yang, and P. Li, "Graph auto-encoder via neighborhood wasserstein reconstruction," in *ICLR*, 2022.

[167] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *NeurIPS*, vol. 26, pp. 2292–2300, 2013.

[168] N. Bonneel, J. Rabin, G. Peyré, and H. Pfister, "Sliced and radon wasserstein barycenters of measures," *J Math Imaging Vis*, vol. 51, no. 1, pp. 22–45, 2015.

[169] T. Kerdoncuff, R. Emonet, and M. Sebban, "Sampled gromov wasserstein," *ML*, vol. 110, no. 8, pp. 2151–2186, 2021.

[170] T. Séjourné, F.-X. Vialard, and G. Peyré, "The unbalanced gromov wasserstein distance: Conic formulation and relaxation," *NeurIPS*, vol. 34, pp. 8766–8779, 2021.

[171] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019, pp. 6023–6032.

[172] X. Lin, K. Chen, and K. Jia, "Object point cloud classification via poly-convolutional architecture search," in *ACM MM*, 2021, pp. 807–815.

[173] F. Poiesi and D. Boscaini, "Learning general and distinctive 3D local deep descriptors for point cloud registration," *TPAMI*, 2022.

[174] S. Shi, C. Guo, L. Jiang, *et al.*, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *CVPR*, 2020.

[175] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *CVPR*, 2021.

[176] M. Xu, Z. Zhou, and Y. Qiao, "Geometry sharing network for 3d point cloud classification and segmentation.," in *AAAI*, 2020, pp. 12 500–12 507.

[177] W. Yuan, T. Khot, D. Held, C. Mertz, and M. Hebert, "Pcn: Point completion network," in *3DV*, 2018.

[178] J.-B. Grill, F. Strub, F. Altché, *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," in *NeurIPS*, 2020.

[179] M. Sarmad, H. J. Lee, *et al.*, "Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion," in *CVPR*, 2019, pp. 5898–5907.

[180] Y. Sun, Y. Wang, Z. Liu, *et al.*, "Pointgrow: Autoregressively learned point cloud generation with self-attention," in *WACV*, 2020, pp. 61–70.

[181] Y. Shi, M. Xu, S. Yuan, and Y. Fang, "Unsupervised deep shape descriptor with point distribution learning," in *CVPR*, 2020, pp. 9353–9362.

[182] K. Hassani and M. Haley, "Unsupervised multi-task feature learning on point clouds," in *ICCV*, 2019, pp. 8160–8171.

[183] B. Du, X. Gao, W. Hu, and X. Li, "Self-contrastive learning with hard negative sampling for self-supervised point cloud learning," in *ACM MM*, 2021, pp. 3133–3142.

[184] O. Poursaeed, T. Jiang, H. Qiao, N. Xu, and V. G. Kim, "Self-supervised learning of point clouds via orientation estimation," in *3DV*, 2020.

[185] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *ECCV*, 2018.

[186] Y. M. Asano, C. Rupprecht, and A. Vedaldi, "Self-labelling via simultaneous clustering and representation learning," in *ICLR*, 2020.

[187] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *NeurIPS*, vol. 33, pp. 9912–9924, 2020.

[188] T. Chen, S. Kornblith, *et al.*, "A simple framework for contrastive learning of visual representations," *ICML*, 2020.

[189] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM TOG*, vol. 38, no. 5, pp. 1–12, 2019.

[190] H. Liu, M. Cai, and Y. J. Lee, "Masked discrimination for self-supervised learning on point clouds," in *ECCV*, Springer, 2022, pp. 657–675.

[191] A. X. Chang, T. Funkhouser, L. Guibas, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[192] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017, pp. 5828–5839.

[193] S. Gugger and J. Howard, "Adamw and super-convergence is now the fastest way to train neural nets," *last accessed*, vol. 19, 2018.

[194] M. Afham, I. Dissanayake, D. Dissanayake, A. Dharmasiri, K. Thilakarathna, and R. Rodrigo, "Crosspoint: Self-supervised cross-modal contrastive learning for 3d point cloud understanding," in *CVPR*, 2022.

[195] J. Hou, B. Graham, M. Nießner, and S. Xie, "Exploring data-efficient 3d scene understanding with contrastive scene contexts," in *CVPR*, 2021, pp. 15 587–15 597.

[196] A. Sharma, O. Grau, and M. Fritz, "Vconv-dae: Deep volumetric shape learning without object labels," in *ECCV*, 2016, pp. 236–250.

[197] L. Yi, V. G. Kim, D. Ceylan, *et al.*, "A scalable active framework for region annotation in 3d shape collections," *ACM TOG*, 2016.

[198] I. Armeni, O. Sener, A. R. Zamir, *et al.*, "3d semantic parsing of large-scale indoor spaces," in *CVPR*, 2016.

[199] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[200] C. Sharma and M. Kaul, "Self-supervised few-shot learning on point clouds," *NeurIPS*, vol. 33, pp. 7212–7221, 2020.

[201] R. Zhou and E. A. Hansen, "Beam-stack search: Integrating backtracking with beam search.," in *ICAPS*, 2005, pp. 90–98.

[202] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 2, 1981, pp. 674–679.

[203] J. Li, S. Huang, H. Cui, Y. Ma, and X. Chen, "Automatic point cloud registration for large outdoor scenes using a priori semantic information," *RS*, vol. 13, no. 17, p. 3474, 2021.

[204] G. Truong, S. Z. Gilani, S. M. S. Islam, and D. Suter, "Fast point cloud registration using semantic segmentation," in *DICTA*, IEEE, 2019, pp. 1–8.

[205] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *IJCV*, vol. 56, no. 3, pp. 221–255, 2004.

[206] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.

[207] Z. Wu, S. Song, A. Khosla, *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.

[208] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in rgb-d images," in *CVPR*, 2013, pp. 2930–2937.

[209] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *TPAMI*, vol. 38, no. 11, pp. 2241–2254, 2015.

[210] X. Li, J. K. Pontes, and S. Lucey, "Pointnetlk revisited," in *CVPR*, 2021, pp. 12 763–12 772.

[211] J. Li, P. Zhao, Q. Hu, and M. Ai, "Robust point cloud registration based on topological and cauchy weighted lq-norm," *ISPRS J. Photogramm. Remote Sens.*, vol. 160, pp. 244–259, 2020.

[212] M. Fey, J. E. Lenssen, C. Morris, J. Masci, and N. M. Kriege, "Deep graph matching consensus," in *ICLR*, 2020.

[213] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *CVPR*, 2018, pp. 2684–2693.

[214] K. Pham, K. Le, N. Ho, T. Pham, and H. Bui, "On unbalanced optimal transport: An analysis of sinkhorn algorithm," in *ICML*, PMLR, 2020, pp. 7673–7682.

[215] A. Iusem and R. D. Monteiro, "On dual convergence of the generalized proximal point method with bregman distances," *Math. Oper. Res.*, vol. 25, no. 4, pp. 606–624, 2000.

[216] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard, "Scaling algorithms for unbalanced optimal transport problems," *Math. Comp.*, vol. 87, no. 314, pp. 2563–2609, 2018.

[217] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3d: A modern library for 3d data processing," *arXiv preprint arXiv:1801.09847*, 2018.

[218] Y. Sun, C. Cheng, Y. Zhang, *et al.*, "Circle loss: A unified perspective of pair similarity optimization," in *CVPR*, 2020, pp. 6398–6407.

[219] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, IEEE, 2012, pp. 3354–3361.

[220] X. Bai and et al., "D3feat: Joint learning of dense detection and description of 3d local features," in *CVPR*, 2020, pp. 6359–6367.

[221] S. Ao, Q. Hu, B. Yang, A. Markham, and Y. Guo, "Spinnet: Learning a general surface descriptor for 3d point cloud registration," in *CVPR*, 2021, pp. 11 753–11 762.

[222] H. Wang, Y. Liu, Z. Dong, and W. Wang, "You only hypothesize once: Point cloud registration with rotation-equivariant descriptors," in *ACM MM*, 2022, pp. 1630–1641.

[223] Z. Sun, R. Zhang, J. Hu, and X. Liu, "Probability re-weighted 3d point cloud registration for missing correspondences," *Multimed. Tools. Appl.*, vol. 81, no. 8, pp. 11 107–11 126, 2022.

[224] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *CVPR*, 2018, pp. 4606–4615.

[225] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Comput Vis Media*, vol. 7, no. 2, pp. 187–199, 2021.

[226] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv*, 2015.

[227] T. Fukunaga and H. Kasai, "Wasserstein k-means with sparse simplex projection," in *ICPR*, IEEE, 2021, pp. 1627–1634.

[228] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[229] P. W. Holland and R. E. Welsch, "Robust regression using iteratively reweighted least-squares," *Commun. Stat. Theory Methods*, vol. 6, no. 9, pp. 813–827, 1977.

[230] A. Handa *et al.*, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *ICRA*, 2014, pp. 1524–1531.

[231] D. Campbell and L. Petersson, "An adaptive data representation for robust point-set registration and merging," in *ICCV*, 2015, pp. 4292–4300.

[232] Kenta-Tanaka, *Probreg*, version 0.1.6, 2019. [Online]. Available: `https://probreg.readthedocs.io/en/latest/`.

[233] X. Huang, S. Li, Y. Zuo, Y. Fang, J. Zhang, and X. Zhao, "Unsupervised point cloud registration by learning unified gaussian mixture models," *RA-L*, 2022.

[234] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *CVPR*, 2015, pp. 5556–5565.