# Rethinking Point Cloud Registration as Masking and Reconstruction

Guangyan Chen[1] Meiling Wang[1] Li Yuan[2] Yi Yang[1] Yufeng Yue[1] *

[1] Beijing Institute of Technology [2] Peking University

## Abstract

*Point cloud registration is essential in computer vision and robotics. In this paper, a critical observation is made that the invisible parts of each point cloud can be directly utilized as inherent masks, and the aligned point cloud pair can be regarded as the reconstruction target. Motivated by this observation, we rethink the point cloud registration problem as a masking and reconstruction task. To this end, a generic and concise auxiliary training network, the Masked Reconstruction Auxiliary Network (MRA), is proposed. The MRA reconstructs the complete point cloud by separately using the encoded features of each point cloud obtained from the backbone, guiding the contextual features in the backbone to capture fine-grained geometric details and the overall structures of point cloud pairs. Unlike recently developed high-performing methods that incorporate specific encoding methods into transformer models, which sacrifice versatility and introduce significant computational complexity during the inference process, our MRA can be easily inserted into other methods to further improve registration accuracy. Additionally, the MRA is detached after training, thereby avoiding extra computational complexity during the inference process. Building upon the MRA, we present a novel transformer-based method, the Masked Reconstruction Transformer (MRT), which achieves both precise and efficient alignment using standard transformers. Extensive experiments conducted on the 3DMatch, ModelNet40, and KITTI datasets demonstrate the superior performance of our MRT over state-of-the-art methods. Codes are available at https://github.com/CGuangyan-BIT/MRA.*

## 1. Introduction

Point cloud registration is a fundamental problem in computer vision and robotics that aims to recover an optimal transformation for point cloud pair alignment. Recent advances in 3D point representation have led to significant attention on learning-based methods [2, 9, 28, 45], such as PointNetLK [2], which utilize neural networks to separately extract point-wise features and establish point-to-point correspondences. Nonetheless, the lack of interaction between point clouds hinders the ability to accurately register partially visible point clouds. Inspired by the recent advances in transformers, transformer-based methods [15, 40, 45, 46] incorporate transformers to exchange information and encode contextual information, exhibiting significant registration accuracy and robustness improvements. However, the limited shared characteristics of low-overlap point cloud pairs produce ambiguity when identifying common structures, thus degrading the performance of such methods.

Recently proposed methods [8, 29, 35, 38] have attempted to address this limitation by introducing dedicated designed encoding methods for measuring pairwise consistency, facilitating the common structure identification and reliable correspondence generation. These methods have been

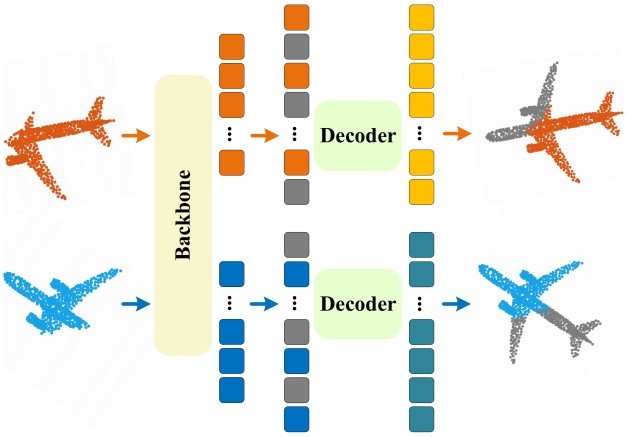


Figure 1. Our MRA considers the invisible parts of each point cloud as inherent masks. During training, the encoded features obtained from the backbone, along with mask tokens, are processed by a decoder that separately constructs the complete point cloud. In this manner, the MRA guides the backbone to capture the geometric details and overall structures. After training, the decoder is detached, thereby avoiding extra inference time.

*Corresponding author: Yufeng Yue (yueyufeng@bit.edu.cn). This work was supported by the National Natural Science Foundation of China under Grant No. NSFC 62233002, 62003039, 61973034, U1913203, the National Key RD Program of China (2022ZD0118), the CAST program under Grant No. YESS20200126.

shown to be effective in low-overlap scenarios. However, such modifications can be limiting constraints in terms of versatility. In addition, these methods incur extra inference time due to their additional encoding calculations.

To mitigate these issues, we revisit the problem of point cloud registration and make a crucial observation: the invisible parts of each point cloud can serve as inherent masks, whereas the aligned point cloud pair can be treated as the reconstruction objective. This observation leads us to exploit the adaptation of the masked data modeling (MDM), which has exhibited remarkable potential in natural language processing (NLP) [11, 24] and computer vision [20, 48]. Inspired by this insight, we redefine the point cloud registration problem as a masking and reconstruction task.

However, adapting MDM to point cloud registration is not straightforward as other point cloud tasks [34, 36]. Firstly, unlike other tasks that only capture relations within a single point cloud, the registration task handles two point clouds. Secondly, the pretraining-tuning paradigm, which is commonly utilized in other tasks, presents difficulties for point cloud registration due to the relatively limited quantity of training data. These differences emphasize the challenges of adapting MDM in point cloud registration, particularly in terms of capturing relations across point clouds and enabling single-shot model training.

Driven by this analysis, we propose a generic plug-and-play training network, termed the Masked Reconstruction Auxiliary Network (MRA). During training, as illustrated in Fig. 1, the MRA separately utilizes the encoded representations of each point cloud obtained from the backbone, to reconstruct the complete point cloud in the coordinate space. After training, the MRA is detached, thus avoiding extra inference time. Unlike previous MDM methods that operate only on visible parts for reconstruction, the MRA takes full advantage of the transformations between point clouds to avoid early leakage of location information. Consequently, it directly utilizes the contextual information obtained from two point clouds, resulting in a concise approach that allows for both inter-point-cloud relation modeling and single-shot model training. Furthermore, this design enables MRA to guide the contextual features in the backbone to capture the geometric details and overall structures of point cloud pairs.

Benefiting from these advantages of our MRA, spatial deviations in the putative corresponding points can be predicted. To this end, the Deviation Correction module is designed to refine the predicted coordinates of the corresponding points. Building upon the proposed modules, we present a novel transformer-based method, Masked Reconstruction Transformer (MRT), which leverages standard transformers and achieves precise and efficient alignment of point clouds. Experiments conducted on various datasets demonstrate that our MRT outperforms state-of-the-art (SOTA) methods and the MRA enhances registration accuracy while avoiding extra inference time. Our main contributions are:

- A novel perspective for rethinking point cloud registration as a masking and reconstruction process is proposed. Based on this perspective, we present the MRT that achieves precise point cloud alignment.
- A versatile plug-and-play training network, the MRA, is developed to guide the backbone by reconstructing the complete point cloud without extra inference time.
- A correspondence prediction module, Deviation Correction, is proposed to compensate for the deviation between the corresponding points.
- Extensive experiments show that our method outperforms the baselines and achieves SOTA performance on the 3DMatch, ModelNet40, and KITTI datasets.

## 2. Related Work

### 2.1. Transformer-based Methods for Registration

Transformers have exhibited great success in NLP [6, 11, 25] and computer vision [13, 31, 42, 53, 54], which has motivated researchers to explore their application in point cloud registration. The deep closest point (DCP) [45] utilizes a dynamic graph CNN (DGCNN) [37] to separately extract features and introduces a transformer [44] to capture the correlations between point clouds. Predator [21] leverages attention mechanisms to conduct information aggregation across a pair of point clouds and predicts overlapping regions for feature sampling purposes, achieving significantly enhanced performance in low-overlap scenarios. CoFiNet [52] extracts features via attention mechanisms in a coarse-to-fine manner and achieves promising performance. The registration transformer (RegTR) [51] utilizes attention layers to directly generate correspondences.

In general, these methods introduce transformers to enable information exchange and encode contextual information, which facilitates the prediction of putative correspondences. However, the limited shared characteristics of the low-overlap point cloud pairs produce obstacles when identifying common structures. To address this issue, several methods have attempted to enhance the ability of networks to capture common structures with dedicated designed encoding modules. Lepard [29] disentangles point cloud representation and utilizes rotary positional encoding [41] to explicitly reveal 3D relative distance information. The geometric transformer [38] calculates pair-wise distances and triplet-wise angles and combines them with self-attention to capture geometric features. Nonetheless, these methods sacrifice versatility and introduce additional computational complexity during the inference process.

### 2.2. Auxiliary Training with Point Clouds

Auxiliary training has been extensively studied in the field of point cloud research, as it can provide a multifold

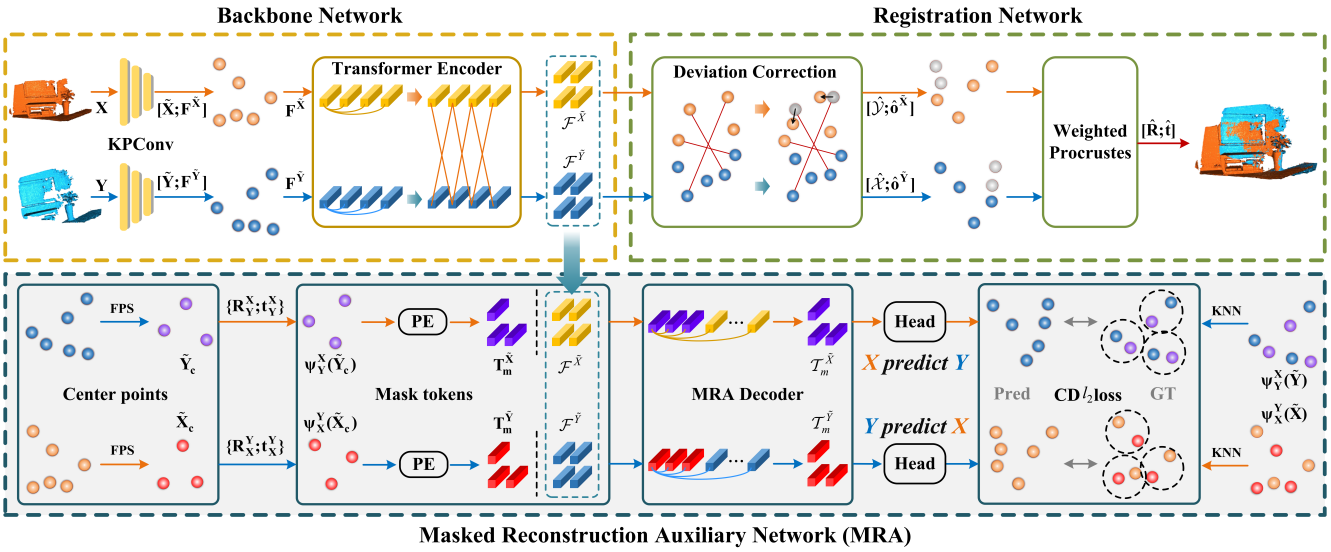


Figure 2. Overview of the MRT. During training, the KPConv extracts features for a sparse set of points. Then, the transformer encoder extracts both intra- and inter-point-cloud features. Afterward, the MRA separately receives the encoded features of each point cloud and predicts the other aligned point cloud, reconstructing the complete point cloud. The point clouds are aligned in the registration network in parallel. During testing, the MRA is detached.

regularization effect during the optimization process while remaining heterogeneous to the main task. SA-SSD [19] introduces an auxiliary network to convert the extracted features back to pointwise representations and then performs foreground segmentation and pointwise center estimation. LabelEnc [18] employs a novel label encoding function to learn latent embeddings from the given ground truth labels, thus providing auxiliary supervision for the training process. LG3D [23] serves as an auxiliary network to achieve enhanced feature learning by obtaining critical representations and fusing object point clouds into the original input point clouds. However, these works focus on single point cloud and are less suitable for registration tasks. In the domain of point cloud registration, DVD [30] additionally utilizes self-reconstruction and normal estimation tasks to consider the intrinsic structural consistency of point clouds. Although DVD also introduces reconstruction tasks, it reconstructs the input point cloud itself, ignoring the relations across point clouds. DeepMapping [12] utilizes deep neural networks (DNNs) as an auxiliary function to model the structure of a scene by estimating the occupancy status of the global coordinates. Different from these methods, we guide the backbone by reformulating the point cloud registration problem as a masking and reconstruction task.

### 2.3. Masked Data Modeling

As a promising self-supervised learning scheme, MDM has achieved promising performance in NLP [11, 24] and computer vision [20, 48]. Data2vec [3] extracts information based on a masked view of the input and predicts contextualized latent representations that contain information from the entire input. The masked autoencoder (MAE) [20] randomly masks patches of the input image. Then, the MAE learns the latent representations from the unmasked patches and reconstructs the missing pixels. Unlike the MAE, SimMIM [48] utilizes a linear layer as its decoder to directly generate the predicted pixels. Following the recent advances in MDM, adaptation has been investigated for use with point clouds. Point-MAE [36] utilizes an asymmetric transformer autoencoder with a shifting mask token operation and learns latent features from the unmasked points to reconstruct the masked points. Voxel-MAE [34] first voxelizes the input point cloud and then predicts the occupancy values of masked voxels instead of the coordinates of the points. However, fundamental differences between point cloud registration and other point cloud processing tasks pose barriers to applying MDM to point cloud registration.

## 3. Masked Reconstruction Transformer

### 3.1. Overall Architecture

Given a source point cloud $\boldsymbol{X} = \{x_1, x_2, ..., x_M\} \subseteq \mathbb{R}^3$ and a target point cloud $\boldsymbol{Y} = \{y_1, y_2, ..., y_N\} \subseteq \mathbb{R}^3$. The objective of point cloud registration is to predict a rotation matrix $\hat{\boldsymbol{R}} \in SO(3)$ and a translation vector $\hat{\boldsymbol{t}} \in \mathbb{R}^3$ that align the source point cloud with the target point cloud.

The overall pipeline of our MRT is depicted in Fig. 2. The training process begins with a backbone that employs the kernel point convolution (KPConv) [43] network to obtain superpoints $\tilde{\boldsymbol{X}}$, $\tilde{\boldsymbol{Y}}$ along with their respective extracted features $\boldsymbol{F}^{\tilde{X}}$, $\boldsymbol{F}^{\tilde{Y}}$. Then, a transformer encoder is employed to learn contextual information and extract features

$\mathcal{F}^{\tilde{X}}$, $\mathcal{F}^{\tilde{Y}}$. These features are separately utilized to reconstruct the complete point cloud using the MRA. In parallel, the encoded features $\mathcal{F}^{\tilde{X}}$, $\mathcal{F}^{\tilde{Y}}$ are utilized to generate the corresponding point clouds $\hat{\mathcal{Y}}$, $\hat{\mathcal{X}}$ and predict overlap scores $\hat{o}^{\tilde{X}}$, $\hat{o}^{\tilde{Y}}$ in the Deviation Correction module. Finally, the weighted Procrustes module estimates the optimal transformation $\{\hat{R}, \hat{t}\}$ based on the predicted correspondences and overlap scores. During testing, the MRA is detached and therefore introduces no extra inference time.

### 3.2. Downsampling and Feature Extraction

The KPConv network consisting of ResNet-like blocks and strided convolutions is utilized for downsampling and feature extraction. The KPConv network downsamples point clouds $\boldsymbol{X} \in \mathbb{R}^{M\times 3}$, $\boldsymbol{Y} \in \mathbb{R}^{N\times 3}$ to obtain superpoints $\tilde{\boldsymbol{X}} \in \mathbb{R}^{M^{'}\times 3}$, $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{N^{'}\times 3}$ and extracts their associated features. Then, the associated features are again projected to obtain features $\boldsymbol{F}^{\tilde{X}} \in \mathbb{R}^{M^{'}\times D}$, $\boldsymbol{F}^{\tilde{Y}} \in \mathbb{R}^{N^{'}\times D}$.

### 3.3. Transformer Encoder

The superpoints $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$, along with their associated features $\boldsymbol{F}^{\tilde{X}}$ and $\boldsymbol{F}^{\tilde{Y}}$, are input into the $L_e$-layer transformer encoder. The transformer encoder conducts information exchange and contextual information extraction, obtaining the encoded features $\mathcal{F}^{\tilde{X}}$ and $\mathcal{F}^{\tilde{Y}}$. Each transformer encoder layer consists of a self-attention sublayer and a cross-attention sublayer, followed by a feedforward network (FFN). To incorporate positional information, sinusoidal positional encodings [44] are added to the inputs of each attention sublayer. Self-attention allows each point to interact with all points within the same point cloud, while cross-attention allows one point cloud to perceive the other.

### 3.4. Masked Reconstruction Auxiliary Network

The idea of MDM is natural and applicable in point cloud registration, as the invisible parts of each point cloud can be directly utilized as inherent masks. Inspired by this observation, we reformulate the problem of point cloud registration as a masking and reconstruction task. Then, the MRA is proposed to reconstruct the complete point cloud. In contrast with previous MDM methods, our MRA fully utilizes the transformations between point clouds to avoid early leakage of location information. It directly employs the encoded features that incorporate the contextual information between two point clouds, allowing for single-shot training and inter-point-cloud relation modeling. This concise design also empowers the MRA to guide the backbone in capturing both geometric details and overall structures.

As illustrated in Fig. 3, our MRA consists of three key components: a point patch generation module, an MRA decoder, and a prediction head. Initially, each point cloud is downsampled and divided into center points and point patches, which are utilized to generate the positional encodings of mask tokens and the reconstruction targets of the MRA, respectively. Subsequently, the MRA decoder separately aggregates the information obtained from the encoded features of each point cloud, and then the prediction head predicts point patches in the other aligned point cloud.

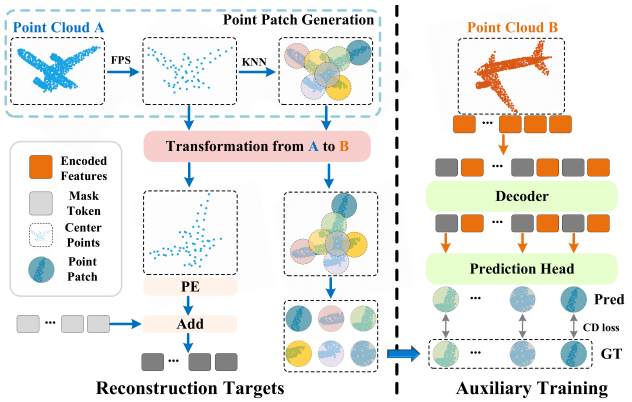


Figure 3. Illustration of the MRA. The input point cloud on the left is first downsampled and divided to obtain its center points and point patches. Then, the transformed center points are utilized to generate the positional encodings of the mask tokens, and the transformed point patches serve as the reconstruction targets for the other point cloud. The network on the right processes the mask tokens and the encoded features to predict the point patches.

**Reconstruction target**. Unlike prior MDM approaches that reconstruct a complete point cloud by only predicting patches in the invisible parts, our method additionally considers the overlapping regions in the other point cloud. This guides the backbone to capture distinctions between the overlapping regions in the two point clouds at the superpoint level and to more accurately model their geometric details. Therefore, our reconstruction target is to recover each point patch in the other aligned point cloud, enabling more precise registration.

**Point patch generation**. A point cloud is a sparse structure that has the property of disorder. Based on this property, the input point clouds are processed by farthest point sampling (FPS) and the K-nearest neighbors (KNN) algorithms to generate center points and point patches. Specifically, for the point cloud $\tilde{\boldsymbol{X}}$, we first sample it into $g$ center points $\tilde{\boldsymbol{X}}_c \in \mathbb{R}^{g\times 3}$ using FPS. Then, the KNN algorithm is utilized to construct $g$ point patches $\tilde{\boldsymbol{X}}_p \in \mathbb{R}^{g\times k\times 3}$ by selecting the $k$ nearest points from $\tilde{\boldsymbol{X}}$ for each center point. In summary, the generation procedure is formulated as:

$$\tilde{\boldsymbol{X}}_c = \mathrm{FPS}(\tilde{\boldsymbol{X}}),\ \tilde{\boldsymbol{X}}_p = \mathrm{KNN}(\tilde{\boldsymbol{X}}_c, \tilde{\boldsymbol{X}}). \tag{1}$$

**MRA decoder**. The MRA decoder is composed of $L_d$ layers, where each decoder layer consists of a self-attention sublayer and a feedforward sublayer. The decoder separately processes full sets of tokens $\boldsymbol{T}_f^{\tilde{X}}$ and $\boldsymbol{T}_f^{\tilde{Y}}$, which compose the encoded features $\mathcal{F}^{\tilde{X}}$ and $\mathcal{F}^{\tilde{Y}}$, as well as the mask

tokens $\boldsymbol{T}_m^{\tilde{X}} \in \mathbb{R}^{g\times D}$ and $\boldsymbol{T}_m^{\tilde{Y}} \in \mathbb{R}^{g\times D}$, respectively. Each mask token is a shared, learned vector that represents a point patch to be predicted.

Due to the order-invariance of the attention mechanism, it is crucial to incorporate positional encodings to specify the point patch that each mask token is responsible for predicting. To this end, sinusoidal positional encodings [44] (PE) are added to the tokens $\boldsymbol{T}_f^{\tilde{X}}, \boldsymbol{T}_f^{\tilde{Y}}$ at each MRA decoder layer, obtaining tokens $\boldsymbol{T}^{\tilde{X}}, \boldsymbol{T}^{\tilde{Y}}$. Specifically, the positional encodings of the mask tokens are generated using the transformed center points of the other point cloud.

$$\boldsymbol{P}_m^{\tilde{X}} = \mathrm{PE}(\psi_Y^X(\tilde{\boldsymbol{Y}}_c))),\ \boldsymbol{P}_m^{\tilde{Y}} = \mathrm{PE}(\psi_X^Y(\tilde{\boldsymbol{X}}_c))), \tag{2}$$

where $\psi_Y^X$, $\psi_X^Y$ are the ground truth transformations from $Y$ to $X$ and from $X$ to $Y$, respectively.

Subsequently, multihead self-attention (MSA) is leveraged to aggregate the information derived from the encoded features. Given the tokens $\boldsymbol{T}^{\tilde{X}}$ as inputs, the MSA operation executes $H$ attention functions Att in parallel. Each Att first generates queries $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$ with projection matrices $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, and $\boldsymbol{W}^V$, respectively:

$$\boldsymbol{Q} = \boldsymbol{T}^{\tilde{X}}\boldsymbol{W}^Q, \quad \boldsymbol{K} = \boldsymbol{T}^{\tilde{X}}\boldsymbol{W}^K, \quad \boldsymbol{V} = \boldsymbol{T}^{\tilde{X}}\boldsymbol{W}^V. \tag{3}$$

Then, each Att obtains an attention map via the scaled dot-product operation and multiplies this map by $V$ to aggregate information. Subsequently, the results of each Att are concatenated and projected with $\boldsymbol{W}^O$ to obtain the final values:

$$\begin{aligned} \boldsymbol{A} &= \mathrm{softmax}(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_K}})\boldsymbol{V}, \\ \mathrm{MSA}(\boldsymbol{T}^{\tilde{X}}) &= \mathrm{Concat}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_H)\boldsymbol{W}^O, \end{aligned} \tag{4}$$

where $d_K$ denotes the dimensionality of the keys $\boldsymbol{K}$. The self-attention mechanism establishes relationships among the tokens in $\boldsymbol{T}^{\tilde{X}}$, thereby enabling the mask tokens $\boldsymbol{T}_m^{\tilde{X}}$ to receive information from the encoded features $\boldsymbol{\mathcal{F}}^{\tilde{X}}$.

In addition to the self-attention sublayer, each MRA decoder layer contains a two-layer fully connected (FC) feed-forward network (FFN) with a rectified linear unit (ReLU) activation function. The FFN is separately and identically applied to each position as follows:

$$\mathrm{FFN}(\boldsymbol{T}) = \mathrm{ReLU}(\boldsymbol{T}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{5}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ are the learnable parameters in the FC network. Through the MRA decoder, the tokens $\boldsymbol{T}^{\tilde{X}}$, $\boldsymbol{T}^{\tilde{Y}}$ are decoded into $\boldsymbol{\mathcal{T}}^{\tilde{X}}$, $\boldsymbol{\mathcal{T}}^{\tilde{Y}}$. Then, the decoded mask tokens $\boldsymbol{\mathcal{T}}_m^{\tilde{X}}$, $\boldsymbol{\mathcal{T}}_m^{\tilde{Y}}$ are fed into the prediction head.

**Prediction head**. The prediction head is utilized to reconstruct each point patch of the other aligned point cloud in the coordinate space. It consists of a two-layer multilayer perceptron (MLP) with two FC layers and ReLU activation. The prediction head projects the decoded mask tokens $\boldsymbol{\mathcal{T}}_m^{\tilde{X}}$ and $\boldsymbol{\mathcal{T}}_m^{\tilde{Y}}$ to vectors, where the number of output channels equals the total number of coordinates in a patch. Then, these vectors are reshaped to construct the predicted point patches $\hat{\boldsymbol{Y}}_p \in \mathbb{R}^{g\times k\times 3}$, $\hat{\boldsymbol{X}}_p \in \mathbb{R}^{g\times k\times 3}$ in a mutual manner:

$$\begin{aligned} \hat{\boldsymbol{Y}}_p &= \mathrm{Reshape}(\mathrm{MLP}(\boldsymbol{\mathcal{T}}_m^{\tilde{X}})), \\ \hat{\boldsymbol{X}}_p &= \mathrm{Reshape}(\mathrm{MLP}(\boldsymbol{\mathcal{T}}_m^{\tilde{Y}})). \end{aligned} \tag{6}$$

**Reconstruction loss**. Each mask token is responsible for predicting the corresponding point patch in the other aligned point cloud, which is specified by its position encoding. Given the ground truth point patches $\boldsymbol{Y}_p^* = \psi_Y^X(\tilde{\boldsymbol{Y}}_p)$ and $\boldsymbol{X}_p^* = \psi_X^Y(\tilde{\boldsymbol{X}}_p)$, as well as the predicted point patches $\hat{\boldsymbol{Y}}_p$ and $\hat{\boldsymbol{X}}_p$, which are obtained from $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$, the reconstruction loss $\mathcal{L}_r$ is formulated using the $l_2$ Chamfer distance (CD) [14]. $\mathcal{L}_r = \mathcal{L}_r^X + \mathcal{L}_r^Y$, where $\mathcal{L}_r^X$ is defined as:

$$\begin{aligned} \mathcal{L}_r^X = &\frac{1}{\left|\hat{\boldsymbol{Y}}_p\right|}\sum_{a\in\hat{\boldsymbol{Y}}_p}\min_{b\in\boldsymbol{Y}_p^*}\|a-b\|_2^2 \\ &+ \frac{1}{\left|\boldsymbol{Y}_p^*\right|}\sum_{b\in\boldsymbol{Y}_p^*}\min_{a\in\hat{\boldsymbol{Y}}_p}\|a-b\|_2^2, \end{aligned} \tag{7}$$

where $\left|\hat{\boldsymbol{Y}}_p\right|$ is the cardinality of the set $\hat{\boldsymbol{Y}}_p$ and $\|a-b\|_2^2$ is the squared error between $a$ and $b$. To facilitate the convergence of the training process, the point patches are represented by normalized coordinates with respect to their center points. For details on other loss functions used to train our MRT, please refer to the Appendix.

### 3.5. Deviation Correction

In parallel with the reconstruction task, the encoded features $\boldsymbol{\mathcal{F}}^{\tilde{X}}$ and $\boldsymbol{\mathcal{F}}^{\tilde{Y}}$ extracted from the encoder are employed to establish correspondences. To achieve efficient registration, the corresponding points $\boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{X}}$ for $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$ are generated between superpoints without upsampling. Specifically, the corresponding points $\boldsymbol{\mathcal{Y}}$ are obtained:

$$\boldsymbol{\mathcal{Y}} = \mathrm{softmax}(\boldsymbol{\mathcal{F}}^{\tilde{X}}\boldsymbol{\mathcal{F}}^{\tilde{Y}^T})\tilde{\boldsymbol{Y}}. \tag{8}$$

However, superpoint matching is sparse and loose, which impedes the accurate prediction of the corresponding points. Benefiting from the fine-grained geometric details of point cloud pairs, especially in the overlapping regions, which are preserved in the contextual features, spatial deviations among the predicted corresponding points can be predicted and rectified. Specifically, the coordinates of the points within each predicted correspondence are concatenated with the encoded features and projected with a 2-layer MLP to obtain the compensation values required to rectify their deviations. As such, the corrected corresponding points $\hat{\boldsymbol{\mathcal{Y}}}$ of $\tilde{\boldsymbol{X}}$ are obtained as

$$\begin{aligned} \boldsymbol{\Phi}^{\tilde{X}} &= \mathrm{Concat}(\boldsymbol{\mathcal{F}}^{\tilde{X}}, \tilde{\boldsymbol{X}}, \boldsymbol{\mathcal{Y}}), \\ \hat{\boldsymbol{\mathcal{Y}}} &= \mathrm{MLP}(\boldsymbol{\Phi}^{\tilde{X}}) + \boldsymbol{\mathcal{Y}}. \end{aligned} \tag{9}$$

In parallel, $\hat{\boldsymbol{\mathcal{X}}}$ is obtained through a similar procedure. Compared to directly predicting coordinates for the corre-

sponding points, utilizing compensation values yield tighter prediction intervals, leading to improved accuracy when predicting corresponding points. Subsequently, the overlap scores $\hat{\boldsymbol{o}}^{\tilde{X}}$ and $\hat{\boldsymbol{o}}^{\tilde{Y}}$, which indicate probabilities of points lying in the overlap regions, are generated by a single FC layer with sigmoid activation:

$$\hat{\boldsymbol{o}}^{\tilde{X}} = \text{Sigmoid}(\text{FC}(\boldsymbol{\mathcal{F}}^{\tilde{X}})),\ \hat{\boldsymbol{o}}^{\tilde{Y}} = \text{Sigmoid}(\text{FC}(\boldsymbol{\mathcal{F}}^{\tilde{Y}})). \quad (10)$$

# 4. Experimental Results

## 4.1. Implementation Details

The numbers of transformer encoder and decoder layers, $L_e$ and $L_d$, are set to 6 and 1, respectively. The numbers of point patches in the 3DMatch, ModelNet, and KITTI datasets are set to 64, 32, and 64, respectively. The MRT is trained using AdamW [32] with an initial learning rate of $1e-4$ and a weight decay of $1e-4$; furthermore, the multistep learning rate (LR) schedule method is utilized. For more implementation details, please refer to the Appendix.

## 4.2. Registration Performance on 3DMatch

**3DMatch**. To demonstrate the real-world point cloud registration performance of our method, experiments are conducted on 3DMatch [55]. The 3DMatch dataset is a real-world registration dataset, in which 46 scenes are designed for training, and the remaining 16 scenes are evenly allocated for validation and testing. The comparison methods are evaluated on both the 3DMatch ($> 30\%$ overlap) [55] and 3DLoMatch ($10\%-30\%$ overlap) [21] benchmarks.

**Comparison methods**. The MRT is compared with the latest approaches RegTR [51], Lepard [29], SC$^2$PCR [8], and GeoTransformer (GeoTR) [38]; furthermore, the comparison methods include 3DSN [17], FCGF [10], CG-SAC [39], D3Feat [4], DGR [9], PCAM [7], OMNet [49], DHVR [26], Predator [21], and CoFiNet [52]. The number of interest points in the correspondence-based methods based on RANSAC is set to 5000.

**Evaluation metrics**. Following [51], the performance of each method is evaluated using the *relative rotation error* RRE (the geodesic distance between the estimated and ground truth rotation matrices), *relative translation error* RTE (the Euclidean distance between the estimated and ground truth translations), and *registration recall* RR (the percentage of successful alignments, which are defined as those with correspondence root mean squared errors RMSEs below 0.2 m). Notably, as the MRT establishes correspondences without performing upsampling, the inlier ratio and feature matching recall are not considered in the results.

The qualitative results are shown in Figs. 4(a-d), and the quantitative comparisons are summarized in Table 1. The results show that our method precisely aligns pairs of real-world point clouds, even at low overlap rates, and outperforms the other methods on both 3DMatch and 3DLoMatch.

Table 1. Performance on the 3DMatch and 3DLoMatch benchmarks. The RRE is given in $^\circ$, the RTE in $m$, and the RR in $\%$. The three best results are highlighted in **red**, **green**, **blue**.

| Method | Reference | 3DMatch | | | 3DLoMatch | | |
|---|---|---|---|---|---|---|---|
| | | RRE | RTE | RR | RRE | RTE | RR |
| 3DSN [17] | CVPR 2019 | 2.19 | 0.071 | 78.4 | 3.52 | 0.103 | 33.0 |
| FCGF [10] | CVPR 2019 | 2.14 | 0.070 | 85.1 | 3.74 | 0.100 | 40.1 |
| CG-SAC [39] | T-GE 2020 | 2.42 | 0.076 | 87.5 | 3.86 | 0.109 | 64.0 |
| D3Feat [4] | CVPR 2020 | 2.16 | 0.067 | 81.6 | 3.36 | 0.103 | 37.2 |
| DGR [9] | CVPR 2020 | 2.10 | 0.067 | 85.3 | 3.95 | 0.113 | 48.7 |
| PCAM [7] | ICCV 2021 | 1.80 | **0.059** | 85.5 | 3.52 | 0.099 | 54.9 |
| OMNet [49] | ICCV 2021 | 4.16 | 0.105 | 35.9 | 7.29 | 0.151 | 8.4 |
| DHVR [26] | ICCV 2021 | 2.25 | 0.078 | 91.9 | 4.97 | 0.123 | 65.4 |
| Predator [21] | CVPR 2021 | 2.02 | 0.064 | 89.0 | 3.04 | 0.093 | 62.5 |
| CoFiNet [52] | Neurips 2021 | 2.44 | 0.067 | 89.3 | 5.44 | 0.155 | 67.5 |
| RegTR [51] | CVPR 2022 | **1.57** | **0.049** | 92.0 | **2.83** | **0.077** | 64.8 |
| Lepard [29] | CVPR 2022 | 2.48 | 0.072 | **93.5** | 4.10 | 0.108 | 69.0 |
| SC$^2$PCR [8] | CVPR 2022 | 2.08 | 0.065 | **93.3** | 3.46 | 0.096 | **69.5** |
| GeoTR [38] | CVPR 2022 | **1.72** | 0.062 | 92.0 | **2.93** | **0.089** | **75.0** |
| Ours | - | **1.32** | **0.043** | **95.1** | **2.49** | **0.072** | **75.4** |

Specifically, the MRT achieves 95.1% RR on the 3DMatch benchmark, exceeding all comparison methods. Even compared to Lepard and SC$^2$PCR, the MRT achieves an improvement of at least 1.6% in RR and reduces the RRE and RTE by 33.8%-46.7%. Additionally, in comparison with GeoTR on 3DLoMatch, the MRT still achieves higher RR values, while reducing the RRE and RTE. These results demonstrate that our method efficiently captures the overall structures of point cloud pairs, enhancing its ability to identify overlapping regions and predict correspondences. Therefore, our method precisely aligns real-world point clouds with superior accuracy and RR.

## 4.3. Registration Performance on ModelNet40

**ModelNet40**. The proposed algorithm and the baseline methods are evaluated on the ModelNet40 [47] dataset. This dataset includes 12,311 meshed models in 40 categories, of which 5,112 samples are used for training, 1,202 samples are used for validation, and 1,266 samples are used for testing. Following [21, 50, 51], the comparison methods are evaluated under two partial overlap settings: ModelNet which has a 73.5% average overlap rate, and ModelLoNet which possesses a 53.6% average overlap rate.

**Comparison methods**. The MRT is compared with the latest approach RegTR [51]; furthermore, the comparison methods also include ICP [5], FGR [56], PointNetLK (PNetLK) [2], DCP-v2 [45], IDAM [27], RPM-Net [50], OMNet [49], and Predator [21]; specifically, Predator samples 450 points in this experiment.

**Evaluation metrics**. The performance of the comparison methods is evaluated in terms of the RRE, the RTE, and the *Chamfer distance* CD between the registered scans.

The qualitative results are shown in Figs. 4(e,f), and the quantitative comparisons are summarized in Table 2. The

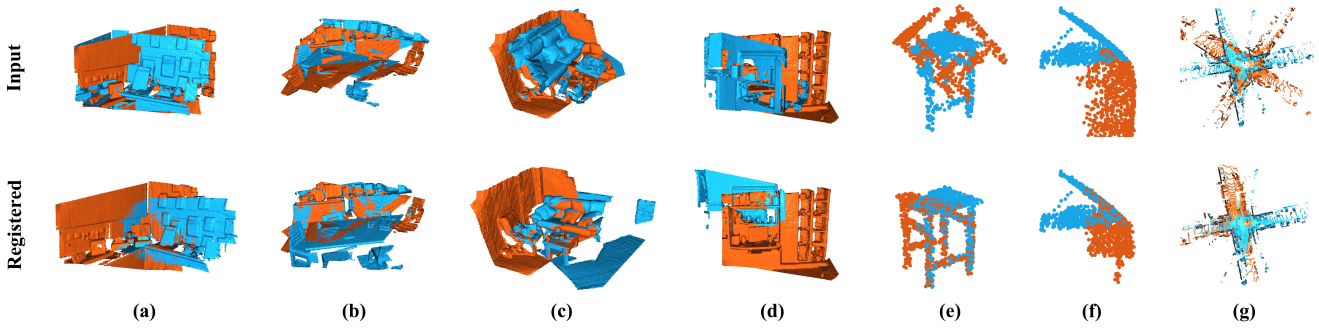


Figure 4. Qualitative registration results for (a, b) 3DMatch, (c, d) 3DLoMatch, (e) ModelNet40, (f) ModelLoNet40 and (g) KITTI.

Table 2. Performance on the ModelNet40 benchmark. The RRE is given in °. The three best results are highlighted in **red**, **green**, and **blue**.

| Method | Reference | ModelNet | | | ModelLoNet | | |
|---|---|---|---|---|---|---|---|
| | | RRE | RTE | CD | RRE | RTE | CD |
| ICP [5] | SPIE 1992 | 27.2 | 0.280 | 0.0230 | 47.5 | 0.479 | 0.0521 |
| FGR [56] | ECCV 2016 | 30.8 | 0.192 | 0.0241 | 58.7 | 0.557 | 0.0517 |
| PNetLK [2] | CVPR 2019 | 29.7 | 0.297 | 0.0235 | 48.5 | 0.507 | 0.0367 |
| DCP-v2 [45] | ICCV 2019 | 11.9 | 0.171 | 0.0117 | 16.5 | 0.300 | 0.0268 |
| RPM-Net [50] | CVPR 2020 | **1.71** | **0.018** | **8.5e-4** | 7.34 | **0.124** | **0.0050** |
| OMNet [49] | ICCV 2021 | 3.12 | 0.037 | 0.0015 | 6.52 | 0.129 | 0.0074 |
| Predator [21] | CVPR 2021 | 1.73 | 0.019 | 8.9e-4 | **5.24** | 0.132 | 0.0083 |
| RegTR [51] | CVPR 2022 | **1.47** | **0.014** | **7.8e-4** | **3.93** | **0.087** | **0.0037** |
| Ours | - | **1.34** | **0.012** | **7.5e-4** | **3.47** | **0.075** | **0.0031** |

results show that our method achieves precise registration on the challenging partially visible point clouds and outperforms the other methods in all metrics. Although RPM-Net [50] additionally utilizes surface normal information, our method achieves superior overall accuracy. Compared with the latest method, RegTR, our method further reduces the rotation and translation errors. The experimental results verify that the MRA enriches the captured geometric details and precisely predicts the spatial deviations between the established correspondences.

### 4.4. Registration Performance on KITTI

**KITTI**. To exhibit the performance of our method on a large-scale point cloud dataset, the MRT and baseline methods are evaluated on the KITTI [16] dataset. The KITTI dataset contains 11 sequences of LiDAR-scanned outdoor driving scenarios, of which 0-5 are used for training, 6-7 are used for validation, and 8-10 are used for testing. Following [21], only point cloud pairs that are at most 10 m away from each other are utilized for evaluation purposes.

**Comparison methods**. The MRT is compared with the latest approaches, SC$^2$PCR [8] and GeoTransformer (GeoTR) [38]; the baseline methods also include RANSAC-based methods: FCGF [10], D3Feat [4], SpinNet [1], Predator [21], and CoFiNet [52]; RANSAC-free methods: FMR [22], DGR [9], and HRegNet [33].

Table 3. Performance on KITTI benchmark. The three best results are highlighted in **red**, **green**, and **blue**.

| Method | Reference | RRE(°) | RTE(m) | RR(%) |
|---|---|---|---|---|
| FCGF [10] | ICCV 2019 | 0.30 | 0.095 | 96.6 |
| FMR [22] | CVPR 2020 | 1.49 | 0.660 | 90.6 |
| DGR [9] | CVPR 2020 | 0.37 | 0.320 | 98.7 |
| D3Feat [4] | CVPR 2020 | 0.30 | 0.072 | **99.8** |
| HRegNet [33] | ICCV 2021 | **0.29** | 0.120 | 99.7 |
| SpinNet [1] | CVPR 2021 | 0.47 | 0.099 | 99.1 |
| Predator [21] | CVPR 2021 | **0.28** | **0.068** | **99.8** |
| CoFiNet [52] | Neurips 2021 | 0.41 | 0.082 | **99.8** |
| SC$^2$PCR [8] | CVPR 2022 | 0.32 | **0.072** | 99.6 |
| GeoTR [38] | CVPR 2022 | **0.24** | **0.068** | **99.8** |
| Ours | - | **0.24** | **0.066** | **99.8** |

**Evaluation metrics**. Following [51], the performance of each method is evaluated using the RRE, RTE, and RR (the percentage of successful alignments, whose RRE and RTE values are below 5° and 2 m, respectively).

The qualitative results are shown in Fig. 4(g), and the quantitative comparisons are summarized in Table 3. The results show that our method achieves the best performance on the KITTI benchmark. The experimental results verify that our method attains an enhanced ability to capture overall structures, thereby improving its registration accuracy.

### 4.5. Ablation Studies

The proposed MRT is evaluated through ablation studies conducted on the 3DMatch and 3DLoMatch benchmarks. Table 4 presents the performance of various MRT variants. Furthermore, the proposed auxiliary network, MRA, is integrated with different registration methods to demonstrate its versatility, the results are displayed in Table 5.

**MRA**. MRT$_{w/o\ MRA}$ is trained without the assistance of the MRA, resulting in reduced RR values of 93.5% and 67.8% on the 3DMatch and 3DLoMatch benchmarks, respectively. To verify the versatility of the MRA, we integrate it into the well-established Predator, GeoTR, and RegTR registration methods and compare their perfor-

Table 4. Ablation results on the 3DMatch and 3DLoMatch benchmarks concerning the effects of the different model components. The RR is given in %, the RRE in °, and the RTE in $m$.

| Method | 3DMatch | | | 3DLoMatch | | |
|---|---|---|---|---|---|---|
| | RR | RRE | RTE | RR | RRE | RTE |
| MRT$_{w/o\ MRA}$ | 93.5 | 1.422 | 0.044 | 67.8 | 2.506 | 0.073 |
| MRT$_{w/\ add_mask}$ | 94.3 | 1.356 | 0.044 | 73.6 | 2.306 | 0.070 |
| MRT$_{w/o\ PE_trans}$ | 94.0 | 1.364 | 0.044 | 71.6 | 2.482 | 0.074 |
| MRT$_{w/o\ overlap}$ | 94.1 | 1.453 | 0.046 | 73.2 | 2.645 | 0.076 |
| MRT$_{w/\ l1_loss}$ | 94.4 | 1.413 | 0.045 | 72.4 | 2.482 | 0.074 |
| MRT$_{w/\ l1l2_loss}$ | 95.1 | 1.437 | 0.046 | 74.3 | 2.433 | 0.073 |
| MRT$_{w/o\ correct}$ | 94.2 | 1.372 | 0.043 | 73.5 | 2.510 | 0.074 |
| MRT$_{w/\ project}$ | 94.5 | 1.444 | 0.045 | 71.0 | 2.641 | 0.076 |
| MRT | **95.1** | **1.324** | **0.043** | **75.4** | **2.488** | **0.072** |

mance with that of their vanilla versions. The results, as shown in Table 5, indicate that the integration of the MRA significantly enhances the overall structure modeling performance of these methods, leading to at least 2% increases in RR on both the 3DMatch and 3DLoMatch benchmarks. Moreover, MRA can improve the accuracy of RegTR by up to 6.2%. These findings demonstrate the adaptability of our proposed MRA and its effectiveness when used in combination with other methods.

**Additional masking**. MRT$_{w/\ add_mask}$ additionally randomly masks a proportion of the point patches prior to utilizing the transformer encoder. Although it achieves encouraging results in terms of the RRE and RTE, RR decreases are observed on 3DMatch and 3DLoMatch with values of 94.3% and 73.6%, respectively. We hypothesize that this can be attributed to an excess emphasis on preserving detailed geometric features, which in turn hinders the network from effectively modeling the overall structures.

**Positional encodings**. MRT$_{w/o\ PE_trans}$ generates the positional encodings of mask tokens with the untransformed center points, leading to a performance reduction, particularly on the 3DLoMatch benchmark. These results confirm the importance of utilizing the transformed center points to guide the network to model the overall structures.

**Reconstruction targets**. MRT$_{w/o\ overlap}$ only predicts the invisible parts of each point cloud, excluding the overlapping regions, which leads to reduced registration accuracy. The results demonstrate that the reconstruction of overlapping regions guides the contextual features to capture the geometric details in the overlapping regions, thereby achieving improved accuracy.

**Reconstruction loss functions**. MRT$_{w/\ l1_loss}$ employs the $l1$ CD loss, while MRT$_{w/\ l1l2_loss}$ utilizes both the $l1$ and $l2$ CD losses. However, both variants lead to low RR values and higher RRE and RTE values, as the $l2$ loss is more effective in guiding the network toward convergence when the discrepancies between the predicted and ground truth values are small.

Table 5. Ablation results on 3DMatch and 3DLoMatch concerning the effect of the MRA. The RR is given in %, the RRE in °, and the RTE in $m$.

| Method | 3DMatch | | | 3DLoMatch | | |
|---|---|---|---|---|---|---|
| | RR | RRE | RTE | RR | RRE | RTE |
| Predator | 89.0 | 2.029 | 0.064 | 62.5 | 3.048 | 0.093 |
| Predator + MRA | **92.3** | **1.857** | **0.063** | **67.3** | **2.830** | **0.086** |
| GeoTR | 92.0 | 1.723 | 0.062 | 75.0 | 2.934 | 0.089 |
| GeoTR + MRA | **94.1** | **1.650** | **0.052** | **77.1** | **2.457** | **0.073** |
| RegTR | 92.0 | 1.567 | 0.049 | 64.8 | 2.827 | 0.077 |
| RegTR + MRA | **94.5** | **1.444** | **0.045** | **71.0** | **2.641** | **0.076** |

**Deviation correction**. MRT$_{w/o\ correct}$ predicts the corresponding points without correcting their spatial deviations and solely relies on the established correspondences, leading to degraded performance. This supports the effectiveness of our deviation correction method. On the other hand, MRT$_{w/\ project}$ uses a 2-layer MLP to directly project the coordinates of the corresponding points, and the low performance of this variant indicates that the projection network can yield improved accuracy by focusing on deviations.

### 4.6. Other Ablation Studies

For the inference time required by the comparison methods on the 3DMatch benchmark and more ablation studies, please refer to the Appendix.

## 5. Conclusion

In this work, we introduce a novel perspective regarding point cloud registration by rethinking it as a masking and reconstruction task. We then propose a generic auxiliary network called the MRA, which predicts the other aligned point cloud and reconstructs the complete point cloud. MRA guides the network to capture the fine-grained geometric details and overall structures of point cloud pairs. Moreover, the MRA can be detached after training to avoid additional computational complexity during inference. Building upon the MRA, a novel transformer-based method MRT is proposed, which achieves both efficient and accurate point cloud alignment. Extensive experiments demonstrate that our proposed MRT achieves SOTA performance. Furthermore, the results indicate that the proposed MRA is a versatile network that can be combined with other methods to further enhance their performance. In the future, we plan to further extend our method to cross-modality (e.g., 2D-3D) registration.

## Acknowledgements

This work was supported in part by the National Key R&D Program of China (2022ZD0118101)

## References


[1] Sheng Ao, Qingyong Hu, Bo Yang, Andrew Markham, and Yulan Guo. Spinnet: Learning a general surface descriptor for 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2021. 7

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019. 1, 6, 7

[3] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555*, 2022. 3

[4] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6359–6367, 2020. 6, 7

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 6, 7

[6] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[7] Anh-Quan Cao, Gilles Puy, Alexandre Boulch, and Renaud Marlet. Pcam: Product of cross-attention matrices for rigid registration of point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13229–13238, 2021. 6

[8] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. Sc2-pcr: A second order spatial compatibility for efficient and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13221–13231, 2022. 1, 6, 7

[9] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2514–2523, 2020. 1, 6, 7

[10] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019. 6, 7

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3

[12] Li Ding and Chen Feng. Deepmapping: Unsupervised map estimation from multiple point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8650–8659, 2019. 3

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[14] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5

[15] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8893–8902, 2021. 1

[16] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 7

[17] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5545–5554, 2019. 6

[18] Miao Hao, Yitao Liu, Xiangyu Zhang, and Jian Sun. Labelenc: A new intermediate supervision method for object detection. In *European Conference on Computer Vision*, pages 529–545. Springer, 2020. 3

[19] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11873–11882, 2020. 3

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 2, 3

[21] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021. 2, 6, 7

[22] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11366–11374, 2020. 7

[23] Yaomin Huang, Xinmei Liu, Yichen Zhu, Zhiyuan Xu, Chaomin Shen, Zhengping Che, Guixu Zhang, Yaxin Peng, Feifei Feng, and Jian Tang. Label-guided auxiliary training improves 3d object detector. In *European Conference on Computer Vision*, pages 684–700. Springer, 2022. 3

[24] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 2, 3

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 2

[26] Junha Lee, Seungwook Kim, Minsu Cho, and Jaesik Park. Deep hough voting for robust global registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15994–16003, 2021. 6

[27] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 378–394. Springer, 2020. 6

[28] Xueqian Li, Jhony Kaesemodel Pontes, and Simon Lucey. Pointnetlk revisited. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12763–12772, 2021. 1

[29] Yang Li and Tatsuya Harada. Lepard: Learning partial point cloud matching in rigid and deformable scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5554–5564, 2022. 1, 2, 6

[30] Dongrui Liu, Chuanchuan Chen, Changqing Xu, Robert Qiu, and Lei Chu. Self-supervised point cloud registration with deep versatile descriptors. *arXiv preprint arXiv:2201.10034*, 2022. 3

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[33] Fan Lu, Guang Chen, Yinlong Liu, Lijun Zhang, Sanqing Qu, Shu Liu, and Rongqi Gu. Hregnet: A hierarchical network for large-scale outdoor lidar point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16014–16023, 2021. 7

[34] Chen Min, Dawei Zhao, Liang Xiao, Yiming Nie, and Bin Dai. Voxel-mae: Masked autoencoders for pre-training large-scale point clouds. *arXiv preprint arXiv:2206.09900*, 2022. 2, 3

[35] Taewon Min, Chonghyuk Song, Eunseok Kim, and Inwook Shim. Distinctiveness oriented positional equilibrium for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5490–5498, 2021. 1

[36] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. *arXiv preprint arXiv:2203.06604*, 2022. 2, 3

[37] Anh Viet Phan, Minh Le Nguyen, Yen Lam Hoang Nguyen, and Lam Thu Bui. Dgcnn: A convolutional neural network over large-scale labeled graphs. *Neural Networks*, 108:533–543, 2018. 2

[38] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11143–11152, 2022. 1, 2, 6, 7

[39] Siwen Quan and Jiaqi Yang. Compatibility-guided sampling consensus for 3-d point cloud registration. *IEEE Transactions on Geoscience and Remote Sensing*, 58(10):7380–7392, 2020. 6

[40] Chenghao Shi, Xieyuanli Chen, Kaihong Huang, Junhao Xiao, Huimin Lu, and Cyrill Stachniss. Keypoint matching for point cloud registration using multiplex dynamic graph attention networks. *IEEE Robotics and Automation Letters*, 2021. 1

[41] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021. 2

[42] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. *arXiv preprint arXiv:2201.02767*, 2022. 2

[43] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 3

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 4, 5

[45] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 1, 2, 6, 7

[46] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240*, 2019. 1

[47] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 6

[48] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 2, 3

[49] Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3132–3141, 2021. 6, 7

[50] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020. 6, 7

[51] Zi Jian Yew and Gim Hee Lee. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of*

*the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6677–6686, 2022. 2, 6, 7

[52] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Advances in Neural Information Processing Systems*, 34:23872–23884, 2021. 2, 6, 7

[53] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 2

[54] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition, 2021. 2

[55] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017. 6

[56] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016. 6, 7