# DiffusionPCR: Diffusion Models for Robust Multi-Step Point Cloud Registration

Zhi Chen[1,2] * Yufan Ren[2] * Tong Zhang[2] Zheng Dang[2]
Wenbing Tao[1,3] Sabine Süsstrunk[2] Mathieu Salzmann[2]
[1]Huazhong University of Science and Technology
[2]School of Computer and Communication Sciences, EPFL [3]Tuke Research

## Abstract

*Point Cloud Registration (PCR) estimates the relative rigid transformation between two point clouds. We propose formulating PCR as a denoising diffusion probabilistic process, mapping noisy transformations to the ground truth. However, using diffusion models for PCR has non-trivial challenges, such as adapting a generative model to a discriminative task and leveraging the estimated nonlinear transformation from the previous step. Instead of training a diffusion model to directly map pure noise to ground truth, we map the predictions of an off-the-shelf PCR model to ground truth. The predictions of off-the-shelf models are often imperfect, especially in challenging cases where the two points clouds have low overlap, and thus could be seen as noisy versions of the real rigid transformation. In addition, we transform the rotation matrix into a spherical linear space for interpolation between samples in the forward process, and convert rigid transformations into auxiliary information to implicitly exploit last-step estimations in the reverse process. As a result, conditioned on time step, the denoising model adapts to the increasing accuracy across steps and refines registrations. Our extensive experiments showcase the effectiveness of our DiffusionPCR, yielding state-of-the-art registration recall rates (95.3%/81.6%) on 3DMatch and 3DLoMatch.*

## 1. Introduction

Point Cloud Registration (PCR) aims to find the optimal rigid transformation between two given point clouds. The classical approach to doing so is the Iterative Closest Point (ICP) algorithm [9], which pairs points across the two point clouds and minimizes their Euclidean distance in coordinate space. However, ICP's vulnerability to local minima, stemming from its non-convex and iterative nature, necessitates precise initialization to avoid getting trapped in suboptimal solutions. While the Branch-and-Bound variants of ICP [10, 55] introduce global optimizations, they suffer from low efficiency and limited applicability in certain scenarios [24]. Recently, learning-based methods, particularly feature-matching techniques, have become predominant in PCR [2, 18, 29, 36, 54, 59, 61]. These methods train neural networks to establish point correspondences [37, 54, 59] and use robust estimators to determine the relative transformation [5, 14, 19, 22].

While learning-based methods achieve impressive performance on several benchmarks, low-overlapping scenarios, as illustrated in Fig. 1, remain a challenge. This is because PCR searches for a transformation that minimizes the average distance between ground-truth point pairs. However, ground-truth pairing information is unknown and thus feature-based methods may erroneously match objects that are not part of the common region, leading to inaccuracies

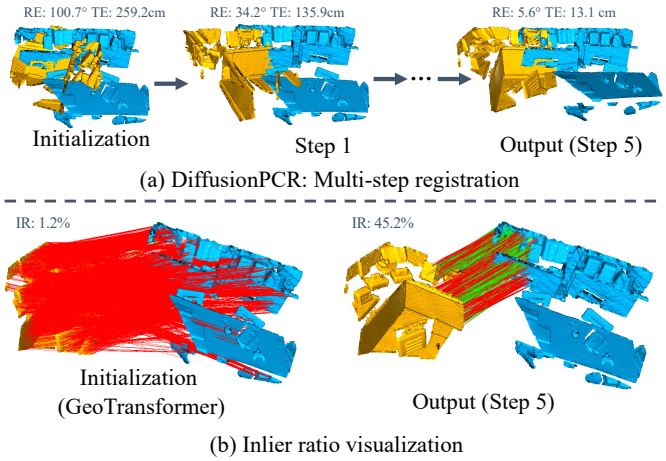


Figure 1. **DiffusionPCR** aligns two point clouds, yellow and blue, depicting partial views of the same scene, with only a small overlapping region (top right). Top row: DiffusionPCR starts with an initialization from a pre-trained network that does not register the point clouds perfectly and refines the result in a multi-step manner. Bottom row: Inlier ratio visualization, where green and red lines indicate correct and incorrect matches, respectively. RE, TE, and IR stand for rotation error (↓), translation error (↓), and inlier ratio (↑), respectively. (Best viewed on a screen when zoomed in).

*Equal contribution

in registration. In Fig. 1, the left wall, chair, and table are not in the overlapping region, and ideally should not be matched with any region.

The recent work PEAL [61] employs a network to refine estimates of pre-trained models using one-way self-attention to focus on predicted overlapping regions. Repeating this process multiple times is feasible, but PEAL's approach does not adjust to the iterative improvements in registration accuracy, leading to suboptimal performance.

Drawing inspiration from the gradual refinement in diffusion models [20, 32, 34, 35, 39], we introduce a denoising diffusion process for multi-step point cloud registration. Yet, applying diffusion models to PCR presents unique challenges. First, the rotation matrix $R \in \mathbb{R}^{3\times3}$ is not in linear space, making it difficult to define interpolation and other operations used in the diffusion process. Second, unlike generative tasks that aim to produce diverse outputs, given two point clouds, a vast portion of the transformation space can be discarded leaving limited room for predictions. Setting diffusion's convergence state to pure random noise makes most training inputs worse than an off-the-shelf PCR model's estimation and learning to map this registration to ground truth is not effective nor efficient. Third, learning-based PCR methods do not take a transformation as input to the network, making it hard to consider the previous step estimates.

To tackle these difficulties, we design DiffusionPCR, a multi-step PCR framework. First, considering the nonlinearity of the rotation matrix, we define rotation in the spherical linear space [46]. Second, to make the network focus on the refinement stage, we train the diffusion model to map an off-the-shelf PCR model's prediction to the ground truth. Finally, in the reverse diffusion process, to make full use of the previous input, we implicitly consider the previous transformation estimate by first converting it to an overlapping region, and propose a novel one-way cross-attention layer to focus on the estimated overlapping regions. Moreover, we use the strategy of ensemble of expert denoisers [6] to boost performance.

Our extensive experiments on three benchmarks demonstrate DiffusionPCR's effectiveness. Our method achieves state-of-the-art registration recall rates (95.3%/81.6%) on the 3DMatch and 3DLoMatch benchmarks. Furthermore, we also show that our method is general enough and can exploit different types of prior distribution beyond the one it is trained on. In summary, the effectiveness of DiffusionPCR comes from the following contributions:

- DiffusionPCR introduces diffusion models to point cloud registration, and we demonstrate their effectiveness.
- We propose a one-way cross-attention mechanism that focuses on the pre-aligned overlapping regions of the two point clouds, resulting in better feature learning.
- We design a diffusion pipeline for PCR, which models rotation in spherical linear space, noise converges to the prior of a pretrained network's estimation, and leverages an ensemble of expert denoisers. DiffusionPCR yields state-of-the-art recall rates on 3DMatch and 3DLoMatch.

## 2. Related Work

**Point Cloud Registration (PCR)** estimates the rigid transformation (rotation and translation) between two point clouds. The most widely used methods for PCR tasks are the Iterative Close Point [9] and its variants, such as ICP-Plane [13], Gen-ICP [43], Go-ICP [55], and GOGMA [10]. The other mainstream solution for PCR is feature-matching-based methods. These methods form correspondences by building and matching point feature descriptors, and then use the robust estimator to remove outliers and recover the relative pose. Traditional descriptors are usually designed by leveraging the local geometric attributes [40, 41] or histograms of spatial distribution [23, 50]. Some learning-based methods [1, 18] design local extractors for enhanced feature representation. Recent advancements, spurred by the Transformers, have led to methods [2, 12, 29, 36, 54, 59, 60] incorporating attention operations into PCR networks for robust results. Additionally, some methods [5, 14, 16, 19, 65] introduce robust estimators for improving alignment from feature correspondences. Besides, researchers also propose deep learning-based end-to-end registration frameworks [3, 15, 53, 57, 58, 62]. Nevertheless, these approaches tend to fail in challenging scenarios such as low-overlapping point clouds. Our work adopts a diffusion model for multi-step registration, augmenting registration with auxiliary information of estimated overlapping regions and adapting the denoising model to different amounts of noise, thus improving registration recall and robustness.

**Using Diffusion Models for Perception Tasks.** Diffusion models are a class of generative models that define the generation process using hierarchical variational autoencoders (HVAE) [28, 33] and achieve remarkable results in image generation [20]. Beyond images, researchers also explored using diffusion models for generating other modalities, such as point clouds [64], voxels [66], and text [31, 56].

The efficacy of diffusion models in generative tasks has inspired many researches to use it for discriminative tasks, such as stereo [45], monocular pose estimation [26, 44], monocular depth estimation [21, 42], camera pose estimation [52], semantic segmentation [8], instance segmentation [27], crowd counting [38, 67], and object detection [11]. However, their application to PCR remains unexplored. We introduce a novel approach that reformulates PCR as a conditional diffusion process on the rigid transformation, conditioned on the two point clouds to be aligned.

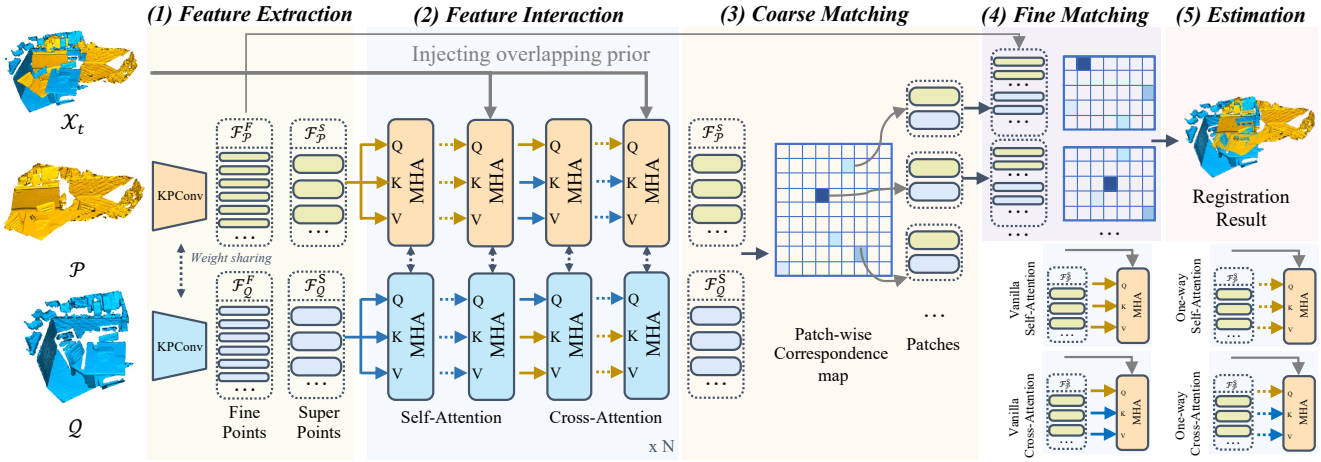


Figure 2. One registration step of DiffusionPCR. (1) For two overlapped point clouds $\mathcal{P}$ and $\mathcal{Q}$, we extract super points ($\mathcal{F}_{\mathcal{P}}^{S}$, $\mathcal{F}_{\mathcal{Q}}^{S}$) and fine points ($\mathcal{F}_{\mathcal{P}}^{F}$, $\mathcal{F}_{\mathcal{Q}}^{F}$) using KPConv. (2) We learn matchable superpoint-wise features using Transformers. Here MHA stands for multi-head attention. Instead of using simple self and cross attention, we leverage the overlapping information from the previous step $\mathcal{X}_t$ using one-way self and cross attention. (3) We calculate the patch-wise correspondence map. (4) We propagate patch-wise correspondences to fine correspondences. (5) We obtain the final estimate using a robust estimator. (Best viewed on a screen when zoomed in)

## 3. Preliminaries

We begin by defining two core concepts: The Point Cloud Registration (PCR) task, and diffusion models.

**Point Cloud Registration (PCR).** PCR is defined as follows: Given two partially overlapped point clouds, denoted as the source point cloud ($\mathcal{P} = \{p_i \in \mathbb{R}^3 | i = 1, ..., N\}$) and the target point cloud ($\mathcal{Q} = \{q_j \in \mathbb{R}^3 | j = 1, ..., M\}$), the objective is to recover the relative rigid transformation, $\mathcal{X} = \{R, t\}$, that aligns their overlapping region.

**Diffusion Models** [28] is a special type of latent variable model where the approximate posterior is fixed to a Markov chain that gradually degrades the data sample $\boldsymbol{x}_0$. In vanilla image diffusion models, Ho et al. [28] define the degradation function as adding Gaussian noise, $N(0, \mathbf{I})$, resulting in a closed-form distribution conditioned on the time step $t$:

$$q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_0\right) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, \left(1 - \bar{\alpha}_t\right)\mathbf{I}\right). \quad (1)$$

The hyper-parameter $\alpha$ is chosen as $\alpha_t^2 = 1 - \sigma_t^2$ to preserve variance. In the reverse process, diffusion models recover the data sample $\boldsymbol{x}_0$ from the noised sample. The model is parameterized by a neural network. To train diffusion models, a simple reconstruction loss is usually employed, where the original weighting term is dropped for empirically better results [28]. This loss is defined as

$$L_{\text{simple}}\left(\theta\right) := \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right]. \quad (2)$$

At inference time, Ho et al. [28] generate samples by executing the reverse process sequentially, resulting in low sampling efficiency. Therefore, DDIM [47] proposes a non-Markovian forward process capable of generating high-quality samples with much fewer sampling steps.

## 4. Conditional Diffusion on Rigid Transformations

We formulate our multi-step point cloud registration as a conditional diffusion process. Following the literature on diffusion models [28, 47], the training stage mainly contains two processes: a forward process and a reverse process. In our context, the forward process gradually degrades the ground-truth rigid transformation to a prior distribution. In the opposite direction, the reverse process takes the noisy transformation as input and recovers the ground-truth alignment. At inference time, we sample a series of gradually decreasing time steps, and infer the result of the current step by inputting the registration of the previous step. In this section, we discuss the details of the forward, reverse, and sampling processes.

### 4.1. Forward Diffusion Process

Following other generative tasks such as image generation, where degradation applies to images, our PCR diffusion model targets degrading the rigid transformation (rotation $R$ and translation $t$). For the degradation function's convergence state, instead of pure Gaussian noise, we opt for an off-the-shelf PCR model's prediction. This strategy simplifies the search towards the ground truth, encouraging the network to focus on refining an initial registration. A similar design choice of constraining the convergence state to be close to ground truth has been studied in the literature for different tasks [11, 45]. The predictions of the off-the-shelf PCR model are imperfect and can be seen as noisy versions of the ground-truth rigid transformations. Second, given the two point clouds and the parameters of the pre-trained network, the prior and ground-truth rigid transformation can

be considered as the Dirac Delta distribution [7]. Then, we can randomly sample an interpolation factor according to the time step and interpolate between the clean data and the noisy prediction to obtain our noisy training data.

Formally, the degradation function in our method is defined as follows:

$$\begin{aligned} \mathbf{R}_t &= Slerp(\mathbf{R}_{\text{quat}}^{\text{prior}}, \mathbf{R}_{\text{quat}}^{\text{gt}}; \sqrt{\alpha_t}) \ , \\ \mathbf{t}_t &= (1-\sqrt{\alpha_t}) \cdot \mathbf{t}^{\text{prior}} + \sqrt{\alpha_t} \cdot \mathbf{t}^{\text{gt}} \ , \end{aligned} \tag{3}$$

where $Slerp(\cdot, \cdot; \cdot)$ is the spherical linear interpolation [46]; $\mathbf{R}_{\text{quat}}^{\text{gt}}$ and $\mathbf{R}_{\text{quat}}^{\text{prior}}$ are the quaternion forms of the prior and ground-truth rotation; $\mathbf{t}^{\text{prior}}$ and $\mathbf{t}^{\text{gt}}$ are the prior and ground-truth translation; $\alpha_t$ is the fusion parameter determined by the time step $t$, which we adopt from [28]. With a smaller $t$, $\alpha_t$ approaches 1, aligning $\mathbf{R}_t, \mathbf{t}_t$ more closely with the ground truth, while a large $t$ shifts $\mathbf{R}_t, \mathbf{t}_t$ towards the prior rigid transformation. The $Slerp$ function ensures a smooth transition in rotation between these states.

### 4.2. Reverse Diffusion Process

**Network backbone.** We adopt GeoTransformer [36] as the backbone of our reverse process, depicted in Fig. 2. Our network consists of five stages. (1) Feature extraction uses the KP-Conv [49], a point-based convolution network to extract features. It gradually down-samples the point cloud in the encoding part to extract sparse points called superpoints with features. It then contains a decoding part, which up-samples the superpoints to get the final fine points. Using the point-to-node strategy [30], each fine point will be allocated to a superpoint, and each superpoint will form a point patch. (2) Feature interaction leverages multi-head attention operations on the two superpoint sets to learn matchable features. (3) Coarse matching computes the correspondence map based on feature similarity, and matches the superpoint by finding the top-k entries. (4) Fine matching performs the fine point matching within the patch correspondences obtained by the coarse matching and gets the fine correspondences. (5) Rigid transformation estimation processes the fine correspondences using a robust estimator, such as RANSAC [22] or LGR [36], to recover the pose.

**Reverse diffusion process.** Different from the vanilla GeoTransformer, which takes as input only two point clouds, our denoising network needs to incorporate the previous estimates. From an optimization perspective, the optimal rigid transformation is obtained by minimizing the average pairwise distance inside the overlapping region, and our intuition is that the overlapping information will help the network to focus on these regions, and leverage this estimated overlapping region to create a more discriminative common feature space.

More specifically, we inject overlapping information via one-way attention operations. One-way attention is similar

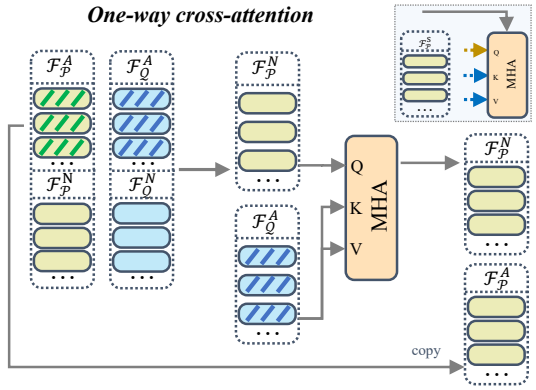


Figure 3. One-way cross-attention. In cross-attention, taking the source point cloud $\mathcal{P}$ as an example, to reduce the influence of non-overlapping regions on the learned features, our one-way cross-attention uses the points in the overlapping region $\mathcal{F}_{\mathcal{Q}}^{A}$ as keys and values of the attention operation. Note that if we set $\mathcal{F}_{\mathcal{P}}^{A}$ as $\mathcal{F}_{\mathcal{Q}}^{A}$, i.e., use self-attention, we get one-way self-attention. (Best viewed on a screen when zoomed in)

to a masking operation, which has been widely used in the Transformer literature [17]. It gives higher weights to specific regions determined by priors. PEAL [61] introduces one-way attention to PCR and augments the features by focusing on the overlapping regions in the point cloud itself. Inspired by this, we propose a more general one-way attention that attends to both intra- and inter-point cloud overlapping regions, arguing that the two overlapping point sets could be complementary. This allows our model to implicitly take the registration result of the last step into account.

To formulate the one-way attention, given two point clouds $\mathcal{P}$, $\mathcal{Q}$ and a transformation estimate $\mathcal{X}_t = \{R_t, t_t\}$, we determine the overlapping points in $\mathcal{P}$ as

$$\mathcal{P}_A^S = \{\mathcal{P}_i^S | \max(O(\mathcal{X}_t(\mathcal{P}_i^S), \mathcal{Q}_j^S)) > 0\}, \mathcal{Q}_j^S \in \mathcal{Q}^S, \tag{4}$$

where $\mathcal{X}_t(\mathcal{P}_i^S)$ transforms the patches corresponding to $\mathcal{P}_i^S$ by $\mathcal{X}_t$, and $O(\cdot, \cdot)$ computes the overlap between two point patches. We call these points anchor points. The anchor points in the target point cloud $\mathcal{Q}_A^S$ are defined similarly. We name the remaining points as non-anchor points, denoted as $\mathcal{P}_N^S$ and $\mathcal{Q}_N^S$.

Then, we perform the attention operation between the anchor points and the non-anchor points, as shown in Fig. 3. Formally, the one-way attention (OA) can be written as

$$\begin{aligned} \text{OA}(\mathcal{F}^A, F^N) &= \mathcal{F}^N + \text{MLP}(F^N + w \times V^A), \\ w &= softmax(Q^N (K^A)^T / \sqrt{D}), \end{aligned} \tag{5}$$

where $\mathcal{F}^A$ and $\mathcal{F}^N$ are the features of the anchor points and the non-anchor points, respectively. Here, we omit the source point and target point subscripts for brevity. $Q^N$, $K^A$, and $V^A$ are produced by applying linear transformations to $\mathcal{F}^A$ and $\mathcal{F}^N$. Instead of taking the anchor points

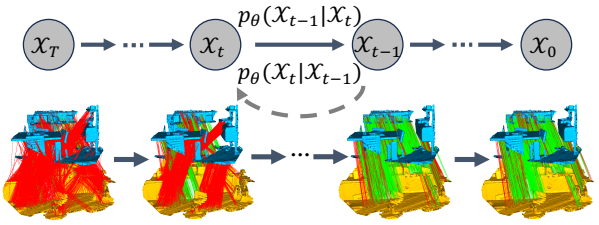


Figure 4. Top row: Graphical model for DiffusionPCR. Bottom row: Illustration of match accuracy of each step. The green and red lines indicate correct and wrong matches, respectively. (Best viewed on a screen when zoomed in)

and non-anchor points from the same point cloud, as in one-way self-attention, we propose to take the anchor points and non-anchor points from the counterpart point cloud, resulting in one-way cross-attention (Fig. 3). To further boost performance, we use an ensemble of denoisers as in [6].

### 4.3. Sampling Process

In the testing phase, as shown in Fig. 4, we sample different time steps and recover the final result gradually. Vanilla diffusion models use DDPM [28], which requires sampling many steps, resulting in low sampling efficiency. Instead, we use DDIM [47], which is capable of generating high-quality samples with much fewer sampling steps. DDIM's inference has a hyper-parameter controlling how much noise is introduced during sampling. We empirically find the zero-noise option, i.e., deterministic sampling, to perform better. We speculate the reason to be that this scheme is more closely aligns with the training phase.

### 4.4. Loss Functions

To train the diffusion model, we adopt the same training losses as GeoTransformer [36], which combines an overlap-aware circle loss $\mathcal{L}_{oc}$ and a point matching loss $\mathcal{L}_p$, i.e.,

$$\mathcal{L} = \mathcal{L}_{oc} + \mathcal{L}_p. \tag{6}$$

The overlap-aware circle loss is an extension of the circle loss [48] that focuses more on the positive samples with high overlap. The point-matching loss is a negative log-likelihood loss on the fine point correspondences.

## 5. Experiments

### 5.1. Implementation Details

For a fair comparison with the baselines, we follow GeoTransformer [37] in our implementation details and experiments. Before feeding the point clouds into our network, we down-sample them using voxel grids, where the voxel size is set as 2.5cm and 30cm for the 3DMatch/3DLoMatch and KITTI benchmarks, respectively. Considering the fact that, on average, the KITTI benchmark has more points in each point cloud than 3DMatch/3DLoMatch, following GeoTransformer, we employ a 5-stage KP-Conv for KITTI, and a 3-stage KP-Conv for 3DMatch/3DLoMatch. We train the KITTI model for 80 epochs and the 3DMatch/3DLoMatch models for 20 epochs, taking around 48h on a single V100 GPU. In the denoising network, we repeat the attention module three times.

### 5.2. Indoor Scenes: 3DMatch & 3DLoMatch

**Datasets.** Following previous works [36, 54, 59], we evaluate our method on the 3DMatch [63] and 3DLoMatch [29] benchmarks. These two datasets are created from 62 RGB-D scenes, of which 46, 8, and 8 scenes are used for training, validation and testing, respectively. A crucial distinction is 3DLoMatch's lower overlap ratio (10% – 30%) compared to 3DMatch's $> 30\%$.

**Metrics.** We adopt Predator's[29] metrics and report Registration Recall (RR), Feature Matching Recall (FMR) and Inlier Ratio (IR) across varying numbers of correspondences. RR measures the percentage of point cloud pairs aligned within an RMSE (Root Mean Square Error), i.e., $\mathrm{RMSE} < 0.2m$). IR represents the ratio of correspondences within a residual threshold under the true transformation, while FMR evaluates the fraction of point cloud pairs whose IR exceeds 5%.

**Registration results.** Tab. 1 compares our method with recent deep learning-based baselines, including 4 local descriptors (FCGF, D3Feat, SpinNet, and YOHO), and 9 Transformer-based methods (REGTR, Predator, CoFiNet, GeoTransformer, OIF-Net RoITR, BUFFER, PEAL, and SIRA-PCR). We follow Predator [29]'s protocol, sampling 5 different numbers of correspondences (5000, 2500, 1000, 500, 250) for each method and evaluate the results. As REGTR and BUFFER directly output the final rigid transformation, they are excluded from correspondence metrics. PEAL can use either 2D or 3D information as prior. While the official 3DMatch and 3DLoMatch datasets do not give the 2D information, PEAL generates the 2D prior by itself. Since the remaining methods do not need the 2D information, in Tab. 1, we report the result of PEAL using a 3D prior as input for a fair comparison. We compare the PEAL-2D with our method in a separate table (Tab. 2). Note that PEAL and our method both adopt iterative optimization, and we use the same number of steps (set to 5) in our experiments.

RR directly reflects the success rate of registration (RMSE smaller than a threshold), and thus usually is considered as the most important metric. As shown in Tab. 1, our method achieves the highest RR by sampling different numbers of correspondences. Our method significantly outperforms all one-pass methods, i.e., the methods that directly use the single-step network to estimate the results, such as our backbone GeoTransformer. Compared with

| | Method | Reference | RR (%) ↑ | | | | | IR (%) ↑ | | | | | FMR (%) ↑ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 |
| 3DMatch | | | | | | | | | | | | | | | | | |
| Descriptor | FCGF | ICCV2019 [18] | 85.1 | 84.7 | 83.3 | 81.6 | 71.4 | 56.8 | 54.1 | 48.7 | 42.5 | 34.1 | 97.4 | 97.3 | 97.0 | 96.7 | 96.6 |
| | D3Feat | CVPR2020 [4] | 81.6 | 84.5 | 83.4 | 82.4 | 77.9 | 39.0 | 38.8 | 40.4 | 41.5 | 41.8 | 95.6 | 95.4 | 94.5 | 94.1 | 93.1 |
| | SpinNet | CVPR2021 [1] | 88.6 | 86.6 | 85.5 | 83.5 | 70.2 | 47.5 | 44.7 | 39.4 | 33.9 | 27.6 | 97.6 | 97.2 | 96.8 | 95.5 | 94.3 |
| | YOHO | ACM MM2022 [51] | 90.8 | 90.3 | 89.1 | 88.6 | 84.5 | 64.4 | 60.7 | 55.7 | 46.4 | 41.2 | 98.2 | 97.6 | 97.5 | 97.7 | 96.0 |
| Transformer-based | REGTR | CVPR2022 [58] | | | 92.0 | | | | | - | | | | | - | | |
| | Predator | CVPR2021 [29] | 89.0 | 89.9 | 90.6 | 88.5 | 86.6 | 58.0 | 58.4 | 57.1 | 54.1 | 49.3 | 96.6 | 96.6 | 96.5 | 96.3 | 96.5 |
| | CoFiNet | NeurIPS2021 [59] | 89.3 | 88.9 | 88.4 | 87.4 | 87.0 | 49.8 | 51.2 | 51.9 | 52.2 | 52.2 | 98.1 | 98.3 | 98.1 | 98.2 | 98.3 |
| | GeoTransformer | CVPR2022 [36] | 92.0 | 91.8 | 91.8 | 91.4 | 91.2 | 71.9 | 75.2 | 76.0 | 82.2 | 85.1 | 97.9 | 97.9 | 97.9 | 97.9 | 97.6 |
| | OIF-Net | NeurIPS2022 [54] | 92.4 | 91.9 | 91.8 | 92.1 | 91.2 | 62.3 | 65.2 | 66.8 | 67.1 | 67.5 | 98.1 | 98.1 | 97.9 | 98.4 | 98.4 |
| | RoITr | CVPR2023 [60] | 91.9 | 91.7 | 91.8 | 91.4 | 91.0 | **82.6** | **82.8** | 83.0 | 83.0 | 83.0 | 98.0 | 98.0 | 97.9 | 98.0 | 97.9 |
| | BUFFER | CVPR2023 [2] | | | 92.9 | | | | | - | | | | | - | | |
| | PEAL | CVPR2023 [61] | **94.4** | 94.1 | 94.1 | 93.9 | 93.4 | 74.8 | 81.3 | 86.0 | 87.9 | 89.2 | **98.5** | **98.6** | **98.6** | **98.7** | **98.7** |
| | SIRA-PCR | ICCV2023 [12] | 93.6 | 93.9 | 93.9 | 92.7 | 92.4 | 70.8 | 78.3 | 83.7 | 85.9 | 87.4 | 98.2 | 98.4 | 98.4 | 98.5 | 98.5 |
| | **DiffusionPCR** | | **94.4** | **94.3** | **94.5** | **94.0** | **93.9** | 75.0 | 81.6 | **86.3** | **88.2** | **89.4** | 98.3 | 98.3 | 98.3 | 98.3 | 98.3 |
| 3DLoMatch | | | | | | | | | | | | | | | | | |
| | Method | | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 | 5000 | 2500 | 1000 | 500 | 250 |
| Descriptor | FCGF | ICCV2019 [18] | 40.1 | 41.7 | 38.2 | 35.4 | 26.8 | 21.4 | 20.0 | 17.2 | 14.8 | 11.6 | 76.6 | 75.4 | 74.2 | 71.7 | 67.3 |
| | D3Feat | CVPR2020 [4] | 37.2 | 42.7 | 46.9 | 43.8 | 39.1 | 13.2 | 13.1 | 14.0 | 14.6 | 15.0 | 67.3 | 66.7 | 67.0 | 66.7 | 66.5 |
| | SpinNet | CVPR2021 [1] | 59.8 | 54.9 | 48.3 | 39.8 | 26.8 | 20.5 | 19.0 | 16.3 | 13.8 | 11.1 | 75.3 | 74.9 | 72.5 | 70.0 | 63.6 |
| | YOHO | ACM MM2022 [51] | 65.2 | 65.5 | 63.2 | 56.5 | 48.0 | 25.9 | 23.3 | 22.6 | 18.2 | 15.0 | 79.4 | 78.1 | 76.3 | 73.8 | 69.1 |
| Transformer-based | REGTR | CVPR2022 [58] | | | 64.8 | | | | | - | | | | | - | | |
| | Predator | CVPR2021 [29] | 59.8 | 61.2 | 62.4 | 60.8 | 58.1 | 26.7 | 28.1 | 28.3 | 27.5 | 25.8 | 78.6 | 77.4 | 76.3 | 75.7 | 75.3 |
| | CoFiNet | NeueIPS2021 [59] | 67.5 | 66.2 | 64.2 | 63.1 | 61.0 | 24.4 | 25.9 | 26.7 | 26.8 | 26.9 | 83.1 | 83.5 | 83.3 | 83.1 | 82.6 |
| | GeoTransformer | CVPR2022 [36] | 75.0 | 74.8 | 74.2 | 74.1 | 73.5 | 43.5 | 45.3 | 46.2 | 52.9 | 57.7 | 88.3 | 88.6 | 88.8 | 88.6 | 88.3 |
| | OIF-Net | NeurIPS2022 [54] | 76.1 | 75.4 | 75.1 | 74.4 | 73.6 | 27.5 | 30.0 | 31.2 | 32.6 | 33.1 | 84.6 | 85.2 | 85.5 | 86.6 | 87.0 |
| | RoITr | CVPR2023 [60] | 74.7 | 74.8 | 74.8 | 74.2 | 73.6 | **54.3** | 54.6 | 55.1 | 55.2 | 55.3 | **89.6** | **89.6** | **89.5** | **89.4** | **89.3** |
| | BUFFER | CVPR2023 [2] | | | 71.8 | | | | | - | | | | | - | | |
| | PEAL | CVPR2023 [61] | 79.2 | 79.0 | 78.8 | 78.5 | 77.9 | 49.1 | 54.1 | 60.5 | 63.6 | 65.0 | 89.1 | 89.2 | 89.0 | 89.0 | 88.8 |
| | SIRA-PCR | ICCV2023 [12] | 73.5 | 73.9 | 73.0 | 73.4 | 71.1 | 43.3 | 49.0 | 55.9 | 58.8 | 60.7 | 88.8 | 89.0 | 88.9 | 88.6 | 87.7 |
| | **DiffusionPCR** | | **80.0** | **80.4** | **79.2** | **78.8** | **78.8** | 49.7 | **55.4** | **61.8** | **64.5** | **66.2** | 86.3 | 85.9 | 86.0 | 86.1 | 85.9 |

Table 1. Results on indoor datasets. The results of the compared methods are taken from their paper. The best scores are in **bold**.

multi-step methods, PEAL, our method still achieves notably better results. We attribute this superiority to diffusion models and our one-way cross attention. Note that, according to the result provided by PEAL, our method with 5 steps (80.4%) is better than PEAL even with 10 steps (78.8%) on 3DLoMatch. Note also that we can interpret the diffusion's forward process as a data augmentation that effectively trains the network to adapt to the prior in different levels of accuracy.

As for IR, we achieve the best performance in most setups and slightly worse when the number of correspondences is 5000. This indicates that our method matches point pairs accurately when it is confident. For FMR, our method is slightly worse than some baselines on 3DLoMatch. Our speculation is that although our method effectively improves the correspondence accuracy of most point clouds, resulting in significantly higher overall IR and RR, in some point cloud pairs where baseline methods also fail to register, our method might decrease IR of these pairs, reflected by slightly worse FMR.

GeoTransformer [37] proposes an LGR estimator to compute the rigid transformation for coarse-to-fine based methods. We compare the performance of our method with GeoTransformer and PEAL when combined with LGR. Here, we also compare the results of using a 2D prior as input, as introduced in PEAL. As shown in Tab. 2, our method still yields the highest registration recall on both 3DMatch and 3DLoMatch, no matter whether with a 2D prior or 3D prior.

Additionally, we compare our method with the SOTA 3D outlier removal method MAC [65]. Different from our method, MAC uses the rotation and translation error, in-

| | 3DMatch | | | 3DLoMatch | | |
|---|---|---|---|---|---|---|
| | RR | FMR | IR | RR | FMR | IR |
| GeoTransformer | 92.5 | 98.2 | 70.9 | 74.0 | 87.1 | 43.5 |
| PEAL-3D | 94.2 | 98.5 | 73.3 | 79.0 | 87.6 | 49.0 |
| **DiffusionPCR-3D** | 94.4 | 98.3 | 73.4 | 80.0 | 87.0 | 49.6 |
| PEAL-2D | 94.3 | **99.0** | 72.4 | 81.2 | **91.7** | 45.0 |
| **DiffusionPCR-2D** | **95.3** | 98.5 | **73.9** | **81.6** | 87.7 | **50.4** |

Table 2. Registration results with the LGR estimator. The best scores are in **bold**.

| | 3DMatch | | | | | 3DLoMatch | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| MAC [65] | 95.7 | 95.7 | 95.2 | 95.3 | 94.6 | 78.9 | 78.7 | 78.2 | 77.7 | 76.6 |
| **DiffusionPCR-3D** | 96.9 | 96.9 | 97.0 | 96.6 | 96.5 | 84.4 | 84.4 | 83.8 | 83.0 | 82.5 |

Table 3. Comparison with the recent MAC baseline. Registration Recall is reported as the evaluation metric.

stead of the RMSE, as the threshold for computing the registration recall. For a fair comparison, we follow the evaluation strategy of MAC to re-compute the registration recall of our method [1], and present the results in Tab. 3. Following the best result they report, the RR of MAC is obtained by combining it with the GeoTransformer to establish the correspondences. Our method obtains remarkable improvements over MAC on both the 3DMatch and 3DLoMatch datasets.

In Fig. 5, we present qualitative results of our method on some point cloud pairs that share extremely low-overlapping areas. As a comparison, we also put the alignment results of our baseline GeoTransformer, PEAL, and the ground truth. Note that DiffusionPCR successfully aligns the point clouds in these challenging settings, while GeoTransformer and PEAL get completely wrong results.

### 5.3. Outdoor Scenes: KITTI Odometry

**Dataset.** KITTI odometry [25] is composed of 11 sequences of driving scenes scanned by LiDAR. We follow [18, 36] and use sequences 0-5 for training, 6-7 for validation and 8-10 for testing. We use the optimized ground-truth poses with ICP and use only point cloud pairs that are at least 10m away for evaluation.

**Metrics.** We follow [29] and evaluate the methods by three metrics: (1) Relative Rotation Error (RRE), which is the geodesic distance between the estimated and ground-truth rotation matrices; (2) Relative Translation Error (RTE), which measures the Euclidean distance between the estimated and ground-truth translation vectors; (3) Registration Recall (RR), which encodes the fraction of point cloud pairs whose RRE and RTE are both below certain thresholds (RRE$< 5°$ and RTE$< 2m$).

[1]We use the evaluation protocol in MAC's official code released in https://github.com/zhangxy0517/3D-Registration-with-Maximal-Cliques

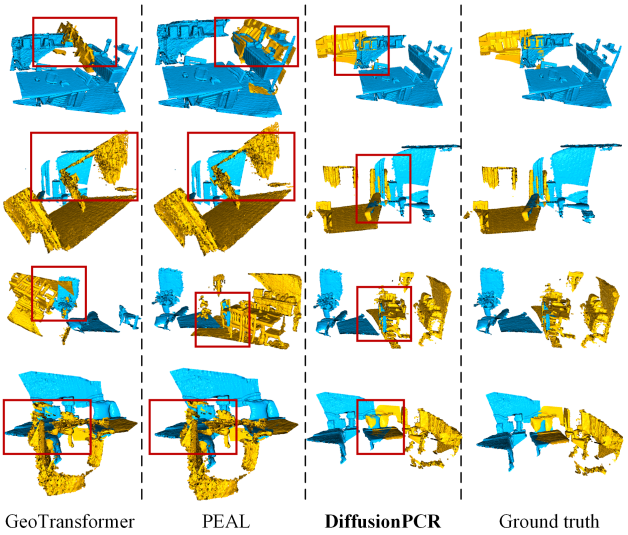


Figure 5. Qualitative results of GeoTransformer, PEAL, and DiffusionPCR compared with the ground truth alignment. The overlapping areas are highlighted by the red boxes. (Best viewed on a screen when zoomed in)

| Methods | Reference | RTE ↓ | RRE ↓ | RR ↑ |
|---|---|---|---|---|
| FCGF | ICCV2019 [18] | 9.5 | 0.30 | 96.6 |
| D3Feat | CVPR2020 [4] | 7.2 | 0.30 | **99.8** |
| SpinNet | CVPR2021 [1] | 9.9 | 0.47 | 99.1 |
| Predator | CVPR2021 [29] | 6.8 | 0.27 | **99.8** |
| CoFiNet | NeurIPS2021 [59] | 8.5 | 0.41 | **99.8** |
| GeoTrans | CVPR2022 [36] | 6.8 | 0.24 | **99.8** |
| OIF-Net | NeurIPS2022 [54] | 6.5 | **0.23** | **99.8** |
| PEAL | CPPR2023 [61] | 6.8 | **0.23** | **99.8** |
| MAC | CVPR2023 [65] | 8.5 | 0.40 | 99.5 |
| **DiffusionPCR** | | **6.3** | **0.23** | **99.8** |

Table 4. Results on KITTI odometry. The best scores are in **bold**.

**Registration results.** We compare our network with 9 recent baselines, including FCGF, D3Feat, SpinNet, Predator, CoFiNet, GeoTransformer, OIF-Net, PEAL, and MAC. PEAL does not have open-source code for the KITTI dataset, so we implemented it by ourselves. As the coarse-to-fine methods, including CoFiNet, GeoTransformer, OIF-Net, PEAL, and ours, can be combined with the LGR estimator [36], we report their results combined with LGR. For the remaining methods, we use RANSAC as their post-processing step. For PEAL and our method, we just perform two steps of iterative optimization because the KITTI dataset is relatively simple. As can be seen in Tab. 4, our method yields 99.8% of RR, which equals the best performance of recent methods. Furthermore, our method achieves the best RRE and RTE compared with all baselines.

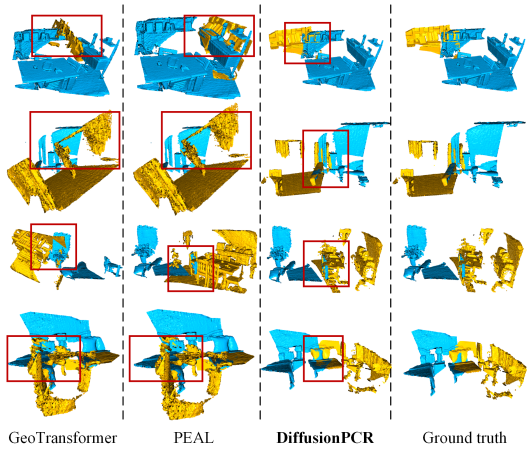


Figure 6. Iteration results using different priors.

### 5.4. Ablation Studies

In Tab. 5, we analyze our design choices by ablating the different components in our framework.

**Initialization with different pretrained models.** In our method, we use the results obtained by the pre-trained GeoTransformer as input. For generalization ability, we use other networks to generate the prior for our method, including Predator [29] and CoFiNet [59]. Furthermore, we also design two control experiments: 1) Not giving prior to our method, but only an identity matrix; 2) Random sampling a rigid transformation from a Gaussian distribution as prior. As shown in Fig. 6, we report the results at different iterative steps. Our method adopts GeoTransformer as the backbone network. Since the Predator and CoFiNet perform worse than GeoTransformer, the performance of using their results as prior is close to using the identity matrix. Through multi-step optimization, even without prior, our method can get near 93.5%/79.0% RR on 3DMatch and 3DLoMatch, respectively, which is much better than the results of GeoTransformer (92.0%/75.0%). This indicates that our method generalizes well to different types of prior. The results of using Gaussian noise as prior is worse than no prior. We speculate the reason is that Gaussian noise introduces uncertainty to the model, which makes the iteration hard to converge.

**Denoising network design.** In our method, the denoising network is designed by utilizing both one-way self-attention and one-way cross-attention. To analyze the effectiveness of these two modules, we remove them separately and summarize the results in Tab. 5. Compared with the baselines, adding either of them boosts performance and we obtain the best results when both of them are applied.

**Degradation schemes.** In our method, we propose a new rigid transformation degradation scheme. Different from the commonly used strategy in generation tasks that relies on random noise as the initial step for training, we interpolate prior information and ground truth in different time steps as the noisy training data for the denoising network. To validate the importance of the degradation scheme, we also use two different degradation schemes to generate the training inputs. The baseline consists of directly using the prior as the input, and a naive strategy is to add noise to the ground truth, as in the vanilla image diffusion models. As shown in Tab. 5, simply adding noise as degradation scheme yields even worse results than the baseline. The proposed deterministic degradation scheme considers the distribution of the results, and (Full) leads to the highest results in all three metrics on both the 3DMatch and 3DLoMatch benchmarks.

| | 3DMatch | | | 3DLoMatch | | |
|---|---|---|---|---|---|---|
| | RR | FMR | IR | RR | FMR | IR |
| Full setting | **94.4** | 98.3 | **73.4** | **80.0** | 87.0 | **49.6** |
| w/o cross | 94.3 | 98.3 | 72.6 | 79.5 | 86.7 | 48.5 |
| w/o self | 93.9 | 98.2 | 72.5 | 79.4 | 86.6 | 48.7 |
| w/o self, cross (baseline) | 92.5 | 98.2 | 70.9 | 74.0 | **87.1** | 43.5 |
| Only prior (baseline) | 93.9 | **98.6** | 70.9 | 78.7 | 86.6 | 47.4 |
| Add noise | 92.7 | 97.3 | 67.6 | 76.9 | 85.8 | 44.5 |

Table 5. Ablations on network components and degradation schemes. The best scores are in **bold**. **Full setting:** The final version of our method. **Only prior**: Directly using the prior as input without degradation scheme. **Add noise**: Using Gaussian noise as degradation function.

## 6. Conclusion

**Conclusion.** In this work, we have presented DiffusionPCR, introducing previously unexplored diffusion models to PCR. Our contributions include a novel degradation scheme that degrades ground truth to prior, using spherical linear space for adding noise, and a novel one-way cross-attention. We demonstrated that our method is general and can use different types of prior distribution beyond the ones it is trained with. Our method achieves state-of-the-art results on the 3DMatch/3DLoMatch and KITTI benchmarks.

**Limitations and future work.** While inspired by image diffusion models, our denoising diffusion process differs from image diffusion, similarly to other works attempting to use diffusion for discriminative tasks [11, 45]. Future work could thus focus on unifying our model with standard image diffusion techniques. In addition, DiffusionPCR has a higher computational cost than the one-pass framework, akin to other iterative frameworks like ICP. A promising future direction could therefore be to use smaller networks and compression strategies such as network pruning to achieve a balance between recall rate and speed.

## 7. Ackownledgement

We appreciate that Dr. Junle Yu and Prof. Wenhui Zhou provide us with the code of PEAL and 2D prior data. Zhi Chen is supported by China Scholarship Council (CSC). This work was supported in part by the Swiss National Science Foundation via the Sinergia grant CRSII5-180359.

## 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 International Conference on Computer Vision*, pages 11753–11762, 2021. 2, 6, 7

[2] Sheng Ao, Qingyong Hu, Hanyun Wang, Kai Xu, and Yulan Guo. Buffer: Balancing accuracy, efficiency, and generalizability in point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1255–1264, 2023. 1, 2, 6

[3] 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. 2

[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] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021. 1, 2

[6] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, Bryan Catanzaro, et al. ediffi: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2, 5

[7] V Balakrishnan. All about the dirac delta function (?). *Resonance*, 8(8):48–58, 2003. 4

[8] Dmitry Baranchuk, Ivan Rubachev, Andrey Voynov, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. *arXiv preprint arXiv:2112.03126*, 2021. 2

[9] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, pages 586–606. Spie, 1992. 1, 2

[10] Dylan Campbell and Lars Petersson. Gogma: Globally-optimal gaussian mixture alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5685–5694, 2016. 1, 2

[11] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 2, 3, 8

[12] Suyi Chen, Hao Xu, Ru Li, Guanghui Liu, Chi-Wing Fu, and Shuaicheng Liu. Sira-pcr: Sim-to-real adaptation for 3d point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14394–14405, 2023. 2, 6

[13] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 2

[14] 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, 2

[15] Zhi Chen, Fan Yang, and Wenbing Tao. Detarnet: Decoupling translation and rotation by siamese network for point cloud registration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 401–409, 2022. 2

[16] Zhi Chen, Kun Sun, Fan Yang, Lin Guo, and Wenbing Tao. Sc2-pcr++: Rethinking the generation and selection for efficient and robust point cloud registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[17] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 4

[18] 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. 1, 2, 6, 7

[19] 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, 2

[20] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[21] Yiqun Duan, Xianda Guo, and Zheng Zhu. Diffusiondepth: Diffusion denoising approach for monocular depth estimation. *arXiv preprint arXiv:2303.05021*, 2023. 2

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1, 4

[23] Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, and Jitendra Malik. Recognizing objects in range data using regional point descriptors. In *European conference on computer vision*, pages 224–237. Springer, 2004. 2

[24] Kexue Fu, Jiazheng Luo, Xiaoyuan Luo, Shaolei Liu, Chenxi Zhang, and Manning Wang. Robust point cloud registration framework based on deep graph matching. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 45(05): 6183–6195, 2023. 1

[25] 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

[26] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qiuhong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13041–13051, 2023. 2

[27] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang. Diffusioninst: Diffusion model for instance segmentation. *arXiv preprint arXiv:2212.02773*, 2022. 2

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 3, 4, 5

[29] 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. 1, 2, 5, 6, 7, 8

[30] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018. 4

[31] Junyi Li, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Renderdiffusion: Text generation as image generation. *arXiv preprint arXiv:2304.12519*, 2023. 2

[32] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11020–11028, 2022. 2

[33] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 2

[34] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2

[35] Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021. 2

[36] 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, 4, 5, 6, 7

[37] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 5, 6

[38] Yasiru Ranasinghe, Nithin Gopalakrishnan Nair, Wele Gedara Chaminda Bandara, and Vishal M Patel. Diffuse-denoise-count: Accurate crowd-counting with diffusion models. *arXiv preprint arXiv:2303.12790*, 2023. 2

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[40] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE, 2008. 2

[41] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009. 2

[42] Saurabh Saxena, Abhishek Kar, Mohammad Norouzi, and David J Fleet. Monocular depth estimation using diffusion models. *arXiv preprint arXiv:2302.14816*, 2023. 2

[43] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, page 435. Seattle, WA, 2009. 2

[44] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. *arXiv preprint arXiv:2303.11579*, 2023. 2

[45] Ruizhi Shao, Zerong Zheng, Hongwen Zhang, Jingxiang Sun, and Yebin Liu. Diffustereo: High quality human reconstruction via diffusion-based stereo using sparse cameras. In *ECCV*, 2022. 2, 3, 8

[46] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 2, 4

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3, 5

[48] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6398–6407, 2020. 5

[49] 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. 4

[50] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010. 2

[51] Haiping Wang, Yuan Liu, Zhen Dong, and Wenping Wang. You only hypothesize once: Point cloud registration with rotation-equivariant descriptors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1630–1641, 2022. 6

[52] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. *arXiv preprint arXiv:2306.15667*, 2023. 2

[53] 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. 2

[54] Fan Yang, Lin Guo, Zhi Chen, and Wenbing Tao. One-inlier is first: Towards efficient position encoding for point

cloud registration. *Advances in Neural Information Processing Systems*, 35:6982–6995, 2022. 1, 2, 5, 6, 7

[55] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015. 1, 2

[56] Yukang Yang, Dongnan Gui, Yuhui Yuan, Haisong Ding, Han Hu, and Kai Chen. Glyphcontrol: Glyph conditional control for visual text generation. *arXiv preprint arXiv:2305.18259*, 2023. 2

[57] 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. 2

[58] 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

[59] 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. 1, 2, 5, 6, 7, 8

[60] Hao Yu, Zheng Qin, Ji Hou, Mahdi Saleh, Dongsheng Li, Benjamin Busam, and Slobodan Ilic. Rotation-invariant transformer for point cloud matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5384–5393, 2023. 2, 6

[61] Junle Yu, Luwei Ren, Yu Zhang, Wenhui Zhou, Lili Lin, and Guojun Dai. Peal: Prior-embedded explicit attention learning for low-overlap point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17702–17711, 2023. 1, 2, 4, 6, 7

[62] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 733–750. Springer, 2020. 2

[63] 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. 5

[64] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 2

[65] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, and Yanning Zhang. 3d registration with maximal cliques. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17745–17754, 2023. 2, 6, 7

[66] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2

[67] Huilin Zhu, Jingling Yuan, Zhengwei Yang, Xian Zhong, and Zheng Wang. Fine-grained fragment diffusion for cross domain crowd counting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5659–5668, 2022. 2