Where Precision Meets Efficiency: Transformation Diffusion Model for Point Cloud Registration

Yongzhe Yuan[1,2], Yue Wu[1,2*], Xiaolong Fan[1,3], Maoguo Gong[1,4], Qiguang Miao[1,2], Wenping Ma[5]

[1]MoE Key Lab of Collaborative Intelligence Systems, Xidian University
[2]School of Computer Science and Technology, Xidian University
[3]School of Electronic Engineering, Xidian University
[4]Academy of Artificial Intelligence, College of Mathematics Science, Inner Mongolia Normal University
[5]School of Artificial Intelligence, Xidian University
{yyz@stu., ywu@, qgmiao@, wpma@mail.}xidian.edu.cn, xiaolongfan@outlook.com, gong@ieee.org

Abstract

We propose a transformation diffusion model for point cloud registration to balance precision and efficiency. Our method formulates point cloud registration as a denoising diffusion process from noisy transformation to object transformation, which is represented by quaternion and translation. Specifically, in training stage, object transformation diffuses from ground-truth transformation to random distribution, and the model learns to reverse this noising process. In sampling stage, the model refines randomly generated transformation to the optimal transformation in a progressive way. We derive the variational bound in closed form for training and provide instantiation of the model. Our diffusion model maps transformation into latent space, and splits the transformation into two components (rotation and translation) based on the fact that they belong to different solution spaces. In addition, our work provides the following crucial findings: (i) Point cloud registration, one of the representative discriminative tasks, can be solved by a generative way and mapped into latent space to obtain new unified probabilistic formulation. (ii) Our model, Transformation Diffusion Model (TDM) can be a plug-and-play agent for point cloud registration, making our method applicable to different deep registration networks. Experimental results on synthetic and real-world datasets demonstrate that, in correspondence-free and correspondence-based scenarios, TDM can both achieve exceeding 60% performance improvements and higher efficiency simultaneously.

Introduction

With the rapid development of 3D data acquisition technology (Guo, Zhu, and Chen 2023), point cloud registration, as a fundamental visual task, plays an crucial role in the 3D vision field, and has been widely applied in various high-level tasks, such as 3D scene reconstruction (Guo et al. 2022), object pose estimation (Jiang et al. 2024), and Simultaneous Localization and Mapping (SLAM) (Zhu et al. 2022). Given two 3D point clouds, the goal of point cloud registration is to find a rigid transformation to align one point cloud to another.

Rigid point cloud registration methods have been evolving in response to the growing complexity of scenarios. These methods have advanced from the most basic technique of utilizing regression to predict transformation parameters in the correspondence-free methods (Huang, Mei, and Zhang 2020; Mei 2021; Mei et al. 2022; Xu et al. 2021), employing correspondence-based methods (Bai et al. 2021; Choy, Dong, and Koltun 2020; Huang et al. 2021; Yew and Lee 2022) and leveraging SVD decomposition to obtain rigid transformations in partially overlapping scenarios. Specifically, in the correspondence-free methods, they are often necessary to seek differences between global features and require features to be sensitive to posture. On the basis of the correspondence-based methods, it is necessary to find the overlapping parts of the point cloud through precise matching and inlier estimation. However, point cloud registration faces a fundamental challenge, ***how to balance precision and efficiency***? Correspondence-free methods tend to prioritize efficiency but exhibit significantly lower precision compared to correspondence-based methods, which leverage larger network structures to enhance performance. On the contrary, employing correspondence-based method enhances precision but this comes at the expense of efficiency.

Inspired by the remarkable achievements of diffusion models in generative AI, we adapt diffusion models to handle the point cloud registration and treat transformation parameters prediction as a generation task. We argue that the process of "noise-to-transformation" in point cloud registration is analogous to the process of "noise-to-point cloud" in point cloud generation, which gradually remove noise from point clouds to generate point clouds with different shapes. Diffusion models have achieved significant success in various generative tasks (Avrahami, Lischinski, and Fried 2022) and recently have applied in some discriminative tasks (Brempong et al. 2022; Graikos et al. 2022; Wu et al. 2023a). However, the application of diffusion models to point cloud registration lags significantly in discriminative tasks and we aim to ulteriorly push forward the development of the point cloud registration pipeline.

Motivated by the discussion above, in this paper, we propose a Transformation Diffusion Model (TDM) for point cloud registration from a novel perspective, intending to simultaneously achieve higher precision and efficiency. Our method formulates point cloud registration as a denoising

*Corresponding author.

diffusion process in $\mathbb{H}$ (quaternion space) and $\mathbb{R}^3$ (translation space), respectively. Noted that the diffusion model parallelly diffuses in two separate spaces based on the fact that they belong to different solution spaces. Starting from purely random transformation (rotation/translation), which do not contain learnable parameters that need to be optimized in training, we expect to gradually refine the predicted transformation until they perfectly align two point clouds. In the training stage, we employ a variance schedule (Ho, Jain, and Abbeel 2020) to control the addition of Gaussian noise to the ground-truth transformation, resulting in noisy transformation. Then we map transformation into latent space and a denoising network is constructed to reverse this noising process. We derive the variational bound in closed form to enable optimize denoising network. In sampling stage, TDM generates transformation by reversing the learned diffusion process, which adjusts a noisy prior distribution to the learned distribution over transformation. Our diffusion model operates in linear space rather than in the $SE(3)$ manifold , firstly due to the need for additional constraint using the Lie algebra $\mathfrak{se}(3)$ associated with $SE(3)$ (Jiang et al. 2024), and secondly because the structure of $SE(3)$ cannot be linearly mapped to the latent space. In addition, our work, the "noise-to-transformation" pipeline, provides the following crucial findings: (i) Point cloud registration, one of the representative discriminative tasks, can be solved by a generative way and mapped into latent space to obtain new unified probabilistic formulation. (ii) Our model, TDM can be a plug-and-play agent for point cloud registration, making our method applicable to different deep registration networks. We only need a small number of sampling steps to improve the registration precision and efficiency.

To validate the effectiveness of the proposed TDM, we comprehensively conduct experiments in both correspondence-based and correspondence-free scenarios on the synthetic dataset ModelNet40 (Wu et al. 2015), real-world dataset 3DMatch (Zeng et al. 2017) and KITTI (Geiger, Lenz, and Urtasun 2012). Experimental results illustrate that TDM can both achieve exceeding 60% performance improvements and higher efficiency simultaneously. To summarize, our contributions are as follows:

- To the best of our knowledge, this is the first study to apply the diffusion model in $\mathbb{H}$ (quaternion space) and $\mathbb{R}^3$ (translation space) and map them into latent space for point cloud registration.
- The customized "noise-to-transformation" pipeline has several appealing advantages, such as simpler the diffusion process and the property of plug-and-play, which allows TDM to be integrated as a hidden pipeline into all existing networks.
- We conduct experiments in both correspondence-based and correspondence-free scenarios. Experimental results illustrate that TDM can both achieve exceeding 60% performance improvements and higher efficiency simultaneously.

Related Works

Point Cloud Registration. Most traditional methods need a good initial transformation and converge to the local minima near the initialization point. One of the most profound methods is the Iterative Closest Point (ICP) (Besl and McKay 1992), which begins with an initial transformation and iteratively alternates between solving two trivial subproblems: finding the closest points as correspondence under current transformation, and computing optimal transformation by SVD (Kurobe et al. 2020). Though ICP can complete a high-precision registration, it is susceptible to the initial perturbation and easily prone to local optima. Thus, variants of ICP have been proposed (Segal, Haehnel, and Thrun 2009; Yang et al. 2015; Fitzgibbon 2003; Rusinkiewicz 2019). However, all these methods retain a few essential drawbacks. Firstly, they depend strongly on the initialization. Secondly, it is difficult to integrate them into the deep learning pipeline as they lack differentiability. Thirdly, explicit estimation of corresponding points leads to quadratic complexity scaling with the number of points (Rusinkiewicz and Levoy 2001), which can introduce significant computational challenges.

To address the aforementioned problems, learning-based methods have made significant advancements in recent years, which are usually divided into correspondence-based methods (Bai et al. 2021; Yuan et al. 2024a; Yew and Lee 2022; Yuan et al. 2024b, 2023) and correspondence-free methods (Huang, Mei, and Zhang 2020; Mei 2021; Mei et al. 2022; Xu et al. 2021). Specifically, in the correspondence-free methods, they are often necessary to seek differences between global features and require them to be sensitive to posture. Such as PointNetLK (Aoki et al. 2019) incorporates the Lucas & Kanade (LK) algorithm the PointNet (Qi et al. 2017) to iteratively align the input point clouds. Correspondence-based methods need to find the correspondences between point clouds through precise matching and inlier estimation. However, point cloud registration faces a fundamental challenge. Correspondence-free methods tend to prioritize efficiency but exhibit significantly lower precision compared to correspondence-based methods, which leverage larger network structures to enhance performance. On the contrary, employing correspondence-based method enhances precision but this comes at the expense of efficiency. In this paper, we aim to push forward the development of the point cloud registration pipeline further with diffusion models to address balancing precision and efficiency.

Diffusion Models. Diffusion models (Ho, Jain, and Abbeel 2020) have emerged as the cutting-edge family of deep generative models, representing a class of highly advanced deep generative models. They have broken the long-time dominance of Generative Adversarial Networks (GANs) (Goodfellow et al. 2020) in the challenging task of image synthesis (Dhariwal and Nichol 2021) and have also shown potential in computer vision (Amit et al. 2021). Specifically, in 3D computer vision, there has been a recent surge of research using generative models for point cloud generation or completion (Luo and Hu 2021). These models are employed to infer missing parts and reconstruct complete shapes and hold significant implications for various downstream tasks, such as 3D reconstruction, augmented reality, and scene understanding (Vahdat et al. 2022). The point diffusion-refinement model (Lyu et al. 2021) is introduced by conditional denoising diffusion probabilistic models to generate a coarse completion from

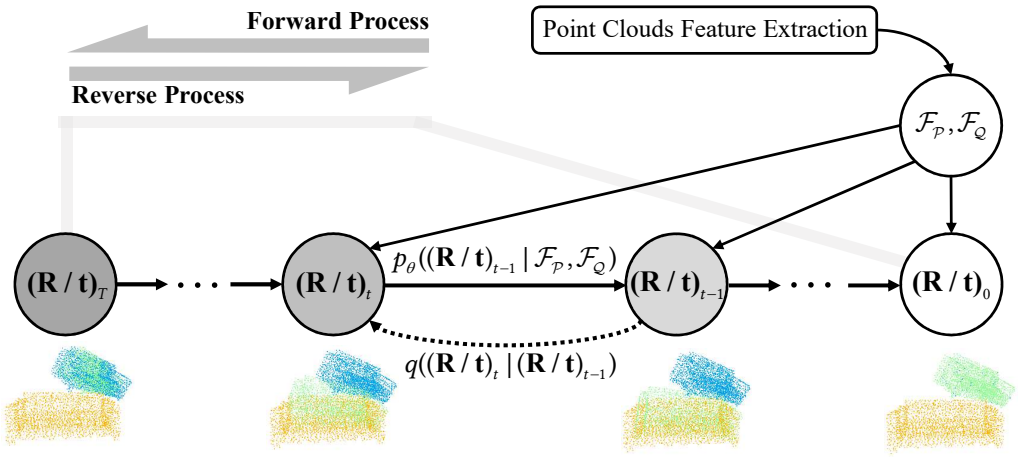

Figure 1: Our transformation diffusion model-based registration framework.

partial observations. Furthermore, this model establishes a point-wise mapping between the generated point cloud and the ground truth. While diffusion models have achieved great success in generation tasks, their potential for discriminative tasks has yet to be fully explored. Currently, there are some pioneering works that apply diffusion models to image segmentation (Wolleb et al. 2022) and object detection (Chen et al. 2023). However, despite the considerable interest in this idea, the application of diffusion models to point cloud registration lags significantly in 3D computer vision.

Transformation Diffusion Model

Point Cloud Registration

Given two point clouds: source point cloud $\mathcal{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1, \dots, N\}$ and template point cloud $\mathcal{Q} = \{\mathbf{q}_j \in \mathbb{R}^3 \mid j = 1, \dots, M\}$, point cloud registration aims to estimate a rigid transformation $\{\mathbf{R}, \mathbf{t}\}$ which accurately aligns $\mathcal{P}$ and $\mathcal{Q}$. The transformation can be solved by:

$$\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbf{p}_i^*, \mathbf{q}_j^*) \in \mathcal{H}^*} \|\mathbf{R} \cdot \mathbf{p}_i^* + \mathbf{t} - \mathbf{q}_j^*\|_2^2, \tag{1}$$

where $\mathcal{H}^*$ is the set of ground-truth correspondences between $\mathcal{P}$ and $\mathcal{Q}$.

In this paper, we split the transformation into two components (rotation and translation) based on the fact that they belong to different solution spaces:

$$\begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{t}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \in SE(3) \rightarrow \begin{cases} [q_1, q_2, q_3, q_4] \in \mathbb{H} \\ [t_1, t_2, t_3] \in \mathbb{R}^3 \end{cases} \tag{2}$$

where q_1, q_2, q_3, q_4 indicates quaternion and t_1, t_2, t_3 indicates translation.

Overview of Architecture

The existing methods are facing bottlenecks in balancing precision and efficiency. TDM aims to address this issue by being a plug-and-play agent that seamlessly integrates into existing networks across various scenarios. In Appendix, we explain how TDM seamlessly integrates into existing network architectures in detail.

Encoder. Encoder takes as input the raw point cloud $\mathcal{P}$ and $\mathcal{Q}$, associated learned features are denoted as $\mathcal{F}_{\mathcal{P}}$ and $\mathcal{F}_{\mathcal{Q}}$, respectively. In correspondence-free and correspondence-based pipeline, encoder extracts global features and point-wise features, respectively. Our implementation follows the encoding schemes of several classical methods, such as PointNet (Qi et al. 2017) and DGCNN (Wang et al. 2019). Distinct from existing point cloud registration methods, we have incorporated an additional **Transformation Encoder** (Fully-Connected layers). Its output is fused with the features of the source point cloud, serving as a channel for diffusion model. We standardize the input of Transformation Encoder as a 7D vectors (include 7 parameters in Equation 2) for ease of encoding, enabling seamless conversion even when the ground truth transformation is in matrix form.

Decoder. To predict transformation, regression and closed-form solution by SVD are employed separately for correspondence-free and correspondence-based pipeline. As mentioned in previous section, TDM operates in $\mathbb{H}$ and $\mathbb{R}^3$ rather than $SE(3)$ manifold. Therefore, for correspondence-based pipeline, we convert the predicted transformation matrix into $\mathbb{H}$ and $\mathbb{R}^3$.

Formulating Point Cloud Registration as Diffusion Model

Inspired by the remarkable achievements of diffusion models, we adopt the concept of denoising diffusion to address the point cloud registration task and propose a transformation diffusion model-based registration framework. Our method formulates point cloud registration as a denoising diffusion process from noisy transformation to object transformation, which is represented by quaternion and translation. Specifically, in training stage, object transformation diffuse from ground-truth transformation to random distribution, and the model learns to reverse this noising process. In sampling stage, the model refines randomly generated transformation to the optimal transformation in a progressive way. Noted that our diffusion process features two critical differences from the conventional one: (i) Our diffusion process operates on the two separate linear spaces parallelly, unlike the diffusion process which acts in $SE(3)$ manifold (Jiang et al.

2024), thus we do not need the additional constraint using the Lie algebra $\mathfrak{se}(3)$ associated with $SE(3)$. (ii) TDM maps transformation into latent space, rather than diffusing independently outside of the network.

TDM includes forward process and reverse process in the training stage, as shown in Figure 1. Starting from a transformation distribution $(\mathbf{R}/\mathbf{t})_0 \sim q\left((\mathbf{R}/\mathbf{t})_0\right)$, forward process q is defined which produces a transformation Markov chain $(\mathbf{R}/\mathbf{t})_1 \rightarrow (\mathbf{R}/\mathbf{t})_2 \rightarrow \ldots \rightarrow (\mathbf{R}/\mathbf{t})_T$ by gradually adding Gaussian noise at each timestep t. In particular, the added noise is controled according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$
\begin{aligned}
q\left((\mathbf{R}/\mathbf{t})_{1:T} \mid (\mathbf{R}/\mathbf{t})_0\right) &= \prod_{t=1}^{T} q\left((\mathbf{R}/\mathbf{t})_t \mid (\mathbf{R}/\mathbf{t})_{t-1}\right), \\
q\left((\mathbf{R}/\mathbf{t})_t \mid (\mathbf{R}/\mathbf{t})_{t-1}\right) &:= \mathcal{N}\left((\mathbf{R}/\mathbf{t})_t; \sqrt{1-\beta_t}(\mathbf{R}/\mathbf{t})_{t-1}, \beta_t \mathbf{I}\right).
\end{aligned} \tag{3}
$$

A notable property of the forward process is that it admits sampling $(\mathbf{R}/\mathbf{t})_t$ at an arbitrary timestep t knowing $(\mathbf{R}/\mathbf{t})_0$ in convenient closed-form evaluation: using the notation $\alpha_t := 1-\beta_t$ and $\tilde{\alpha}_t := \prod_{s=1}^{t} \alpha_s$, we have

$$
\begin{aligned}
q\left((\mathbf{R}/\mathbf{t})_t \mid (\mathbf{R}/\mathbf{t})_0\right) &:= \mathcal{N}\left((\mathbf{R}/\mathbf{t})_t; \sqrt{\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_0, (1-\tilde{\alpha}_t)\mathbf{I}\right) \\
&= \sqrt{\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_0 + \epsilon\sqrt{1-\tilde{\alpha}_t}, \epsilon \sim \mathcal{N}(0, \mathbf{I}).
\end{aligned} \tag{4}
$$

The forward process variances β_t can be learned by reparameterization (Kingma and Welling 2013), or held constant as hyperparameters. In this work, we fix the forward process variances β_t to constant.

Furthermore, reverse process $p_\theta((\mathbf{R}/\mathbf{t})_{0:T} \mid \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q})$ is defined as a reverse transformation Markov chain $(\mathbf{R}/\mathbf{t})_T \rightarrow (\mathbf{R}/\mathbf{t})_{T-1} \rightarrow \ldots \rightarrow (\mathbf{R}/\mathbf{t})_0$ starting at a standard Gaussian prior $p\left((\mathbf{R}/\mathbf{t})_T\right) := \mathcal{N}\left((\mathbf{R}/\mathbf{t})_T; \mathbf{0}, \mathbf{I}\right)$ with a learned denoising network given the learned features of $\mathcal{F}_\mathcal{P}$ and $\mathcal{F}_\mathcal{Q}$:

$$
\begin{aligned}
&p_\theta((\mathbf{R}/\mathbf{t})_{0:T} \mid \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}) = \\
&\qquad p((\mathbf{R}/\mathbf{t})_T) \prod_{t=1}^{T} p_\theta((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}), \\
&p_\theta\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}\right) := \\
&\qquad \mathcal{N}\left((\mathbf{R}/\mathbf{t})_{t-1}; \boldsymbol{\mu}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right), \boldsymbol{\Sigma}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right)\right),
\end{aligned} \tag{5}
$$

which aims to invert the noise corruption process. Noted that we map transformation into latent space and use it as a condition to guide the reverse process. Since calculating $q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t\right)$ (ground truth reverse process) exactly depend on the entire transformation distribution, we approximate $q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t\right)$ using a neural network f_θ with parameter θ, which is optimized to predict a mean $\boldsymbol{\mu}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right)$ and a diagonal covariance matrix $\boldsymbol{\Sigma}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right)$. Intuitively, the forward process $q\left((\mathbf{R}/\mathbf{t})_t \mid (\mathbf{R}/\mathbf{t})_{t-1}\right)$ can be seen as gradually injecting more random noise to the transformation, with the reverse process $p_\theta\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}\right)$ learning to progressively remove noise to obtain realistic transformation by mimicking the ground truth reverse process $q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t\right)$.

Finally, the variational lower bound of the log-likelihood is maximized over the training data, and derived as the optimization objective for training the denoising network f_θ in $\mathbb{H}$ and $\mathbb{R}^3$:

$$
\begin{aligned}
&\mathbb{E}_{q((\mathbf{R}/\mathbf{t})_0)}\left[-\log((\mathbf{R}/\mathbf{t})_0 \mid \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q})\right] = \\
&-\mathbb{E}_{q((\mathbf{R}/\mathbf{t})_0)} \log\left[\int p_\theta((\mathbf{R}/\mathbf{t})_{0:T} \mid \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}) d(\mathbf{R}/\mathbf{t})_{1:T}\right] \\
&\leq \mathbb{E}_{q((\mathbf{R}/\mathbf{t})_{0:T})}\Bigg[\underbrace{D_{\mathrm{KL}}\left(q\left((\mathbf{R}/\mathbf{t})_T \mid (\mathbf{R}/\mathbf{t})_0\right) \| p\left((\mathbf{R}/\mathbf{t})_T\right)\right)}_{①} - \\
&\qquad \underbrace{\log p_\theta\left((\mathbf{R}/\mathbf{t})_0 \mid (\mathbf{R}/\mathbf{t})_1, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}\right)}_{②} + \\
&\qquad \sum_{t>1} D_{\mathrm{KL}}\underbrace{\left(q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, (\mathbf{R}/\mathbf{t})_0\right)\right.}_{③} \| \\
&\qquad \underbrace{\left. p_\theta\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}\right)\right)}_{④}\Bigg],
\end{aligned} \tag{6}
$$

where the inequality is by Jensen's inequality and the Bayes' rule (Sohl-Dickstein et al. 2015).

In the final objective of Equation 6, as the forward process is fixed and $p((\mathbf{R}/\mathbf{t})_T)$ is defined as a Gaussian prior, ① does not affect the learning of θ. ② can be regarded as the reconstruction of the original transformation, which can be computed by estimating $\mathcal{N}\left((\mathbf{R}/\mathbf{t})_0; \boldsymbol{\mu}_\theta\left((\mathbf{R}/\mathbf{t})_1, 1\right), \boldsymbol{\Sigma}_\theta\left((\mathbf{R}/\mathbf{t})_1, 1\right)\right)$ and constructing a discrete decoder. Therefore, the ultimate optimization objective is ③ and ④. Based on Bayes' theorem, we can treat the posterior ③ : $q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, (\mathbf{R}/\mathbf{t})_0\right)$ as the Gaussian distribution:

$$
\begin{aligned}
&q\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, (\mathbf{R}/\mathbf{t})_0\right) := \\
&\qquad \mathcal{N}\left((\mathbf{R}/\mathbf{t})_{t-1}; \tilde{\mu}\left((\mathbf{R}/\mathbf{t})_t, (\mathbf{R}/\mathbf{t})_0\right), \tilde{\beta}_t \mathbf{I}\right),
\end{aligned} \tag{7}
$$

where

$$
\begin{aligned}
\tilde{\mu}\left((\mathbf{R}/\mathbf{t})_t, (\mathbf{R}/\mathbf{t})_0\right) &= \frac{\sqrt{\tilde{\alpha}_{t-1}}\beta_t}{1-\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_0 + \frac{\sqrt{\alpha_t}\left(1-\tilde{\alpha}_{t-1}\right)}{1-\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_t, \\
\tilde{\beta}_t &= \frac{1-\tilde{\alpha}_{t-1}}{1-\tilde{\alpha}_t}\beta_t.
\end{aligned} \tag{8}
$$

How to choice ④ : $p_\theta\left((\mathbf{R}/\mathbf{t})_{t-1} \mid (\mathbf{R}/\mathbf{t})_t, \mathcal{F}_\mathcal{P}, \mathcal{F}_\mathcal{Q}\right)$ is a crucial question by Equation 5. In this paper, we set $\boldsymbol{\Sigma}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right) = \sigma_t^2 \mathbf{I}$, where $\sigma_t^2 = \tilde{\beta}_t$, and $\boldsymbol{\mu}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right)$ is parameterized:

$$
\boldsymbol{\mu}_\theta\left((\mathbf{R}/\mathbf{t})_t, t\right) = \tilde{\boldsymbol{\mu}}_t\left((\mathbf{R}/\mathbf{t})_t, \frac{(\mathbf{R}/\mathbf{t})_t - \sqrt{\tilde{\alpha}_t} f_\theta(\mathcal{P}, \mathcal{Q}, \mathbf{R}_t, \mathbf{t}_t, t)}{\sqrt{1-\tilde{\alpha}_t}}\right), \tag{9}
$$

where $f_\theta(\mathcal{P}, \mathcal{Q}, \mathbf{R}_t, \mathbf{t}_t, t)$ is the output of denoising network. Noted that this parameterization method predicts $(\mathbf{R}/\mathbf{t})_0$ rather than noise. This is a novel configuration designed specifically for the characteristics of network and point cloud registration, as predicting $(\mathbf{R}/\mathbf{t})_0$ holds practical significance.

Algorithm 1: Training Stage

while converged **do**
 # Diffusion Preparation
Sample $(\mathbf{R}/\mathbf{t})_0 \sim q\left((\mathbf{R}/\mathbf{t})_0\right)$
Sample $t\sim$ Uniform$(\{1,\ldots,T\})$
Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0,\mathbf{I})$
Calculate $(\mathbf{R}/\mathbf{t})_t = \sqrt{\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_0 + \epsilon\sqrt{1-\tilde{\alpha}_t}$
 # Registration by $f_\theta(\mathcal{P},\mathcal{Q},\mathbf{R}_t,\mathbf{t}_t,t)$
$\mathcal{F}_\mathcal{P} =$ **Encoder**$(\mathcal{P})$
$\mathcal{F}_\mathcal{Q} =$ **Encoder**$(\mathcal{Q})$
$\mathcal{F}_\mathcal{G} =$ **Transformation Encoder**$(\mathbf{R}_t,\mathbf{t}_t)$
$(\mathbf{R}/\mathbf{t})_{pred} =$ **Decoder**$(\mathcal{F}_\mathcal{P},\mathcal{F}_\mathcal{Q},\mathcal{F}_\mathcal{G})$
if Correspondence-Based **then**
$SE(3) \to \mathbb{H}$ and $\mathbb{R}^3$
end if
Optimize $\nabla_\theta(\|(\mathbf{R}/\mathbf{t})_0 - (\mathbf{R}/\mathbf{t})_{pred}\|^2 + \mathcal{L}_{signal})$
end while

Algorithm 2: Sampling Stage

 # Registration by $f_\theta(\mathcal{P},\mathcal{Q},\mathbf{R}_t,\mathbf{t}_t,t)$
$\mathcal{F}_\mathcal{P} =$ **Encoder**$(\mathcal{P})$
$\mathcal{F}_\mathcal{Q} =$ **Encoder**$(\mathcal{Q})$
Sample $(\mathbf{R}/\mathbf{t})_T \sim \mathcal{N}(0,\mathbf{I})$
for $t = T-1,\ldots,1$ **do**
$\mathcal{F}_\mathcal{G} =$ **Transformation Encoder**$(\mathbf{R}_t/\mathbf{t}_t)$
$(\mathbf{R}/\mathbf{t})_{pred} =$ **Decoder**$(\mathcal{F}_\mathcal{P},\mathcal{F}_\mathcal{Q},\mathcal{F}_\mathcal{G})$
if Correspondence-Based **then**
$SE(3) \to \mathbb{H}$ and $\mathbb{R}^3$
end if
 # Diffusion Process
$(\mathbf{R}/\mathbf{t})_{t-1} = \sqrt{\tilde{\alpha}_{t-1}}(\mathbf{R}/\mathbf{t})_{pred} + \sigma_t\mathbf{z} +$
$\sqrt{1-\tilde{\alpha}_{t-1}-\sigma_t^2}\frac{(\mathbf{R}/\mathbf{t})_t - \sqrt{\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_{pred}}{\sqrt{1-\tilde{\alpha}_t}}$,
$\mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$
end for
return $(\mathbf{R}/\mathbf{t})_0$

Finally, the training objective in training stage is changed as follow:

$$\left\|(\mathbf{R}/\mathbf{t})_0 - f_\theta\left(\sqrt{\tilde{\alpha}_t}(\mathbf{R}/\mathbf{t})_0 + \sqrt{1-\tilde{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2. \quad (10)$$

This objective is tailored to point cloud registration task, guiding the refinement of transformation. Based on the aforementioned process, the sampling procedure is calculated by DDIM (Song, Meng, and Ermon 2020) as follows:

$$(\mathbf{R}/\mathbf{t})_{t-1} = \sqrt{\tilde{\alpha}_{t-1}}f_\theta(\mathcal{P},\mathcal{Q},\mathbf{R}_t,\mathbf{t}_t,t) + \sigma_t\mathbf{z} + \sqrt{1-\tilde{\alpha}_{t-1}-\sigma_t^2}\frac{(\mathbf{R}/\mathbf{t})_t - \sqrt{\tilde{\alpha}_t}f_\theta(\mathcal{P},\mathcal{Q},\mathbf{R}_t,\mathbf{t}_t,t)}{\sqrt{1-\tilde{\alpha}_t}}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \quad (11)$$

Summary

Training Stage. We construct the diffusion process from ground-truth transformation to noisy transformation and then train the model to reverse this process. Algorithm 1 provides the process of TDM training procedure. We add Gaussian noises to the ground truth transformation by $q\left((\mathbf{R}/\mathbf{t})_t \mid (\mathbf{R}/\mathbf{t})_0\right)$. The noise scale is controlled by α_t, which adopts the monotonically decreasing cosine schedule for α_t in different time step t, as proposed in (Nichol and Dhariwal 2021). Noted that we uniformly transform the initial input ground truth transformation matrix $(\mathbf{R}/\mathbf{t})_0$ into 7D vectors (composed of a rotation quaternion and a translation) before encoding, instead of directly utilizing the transformation matrix. This strategy offers several advantages. Firstly, it ensures linearity during diffusion process. Secondly, it reduces the complexity required for the design of the Transformation Encoder network. Note that point cloud registration is not a purely generative task. In addition to Equation 10, optimization needs to be performed in conjunction with the loss specific to the task itself $\mathcal{L}_{signal}$. Due to the plug-and-play nature of TDM, $\mathcal{L}_{signal}$ can represent the original loss function of the embedded network and does not need to be modified.

Sampling Stage. The inference procedure of TDM is a denoising sampling process from noise to object transformation $(\mathbf{R}/\mathbf{t})_0$. Starting from transformation sampled in Gaussian distribution, the model progressively refines its predictions, as shown in Algorithm 2. In sampling stage, data sample $(\mathbf{R}/\mathbf{t})_0$ is reconstructed from noise $(\mathbf{R}/\mathbf{t})_T$ with the model $f_\theta(\mathcal{P},\mathcal{Q},\mathbf{R}_t,\mathbf{t}_t,t)$. The transformation from the last sampling step are sent into the transformation encoder again until a finite number of iterations have been completed.

Experiments

Setup

We evaluate the proposed method on synthetic dataset ModelNet40 (Wu et al. 2015), real-world dataset 3DMatch (Zeng et al. 2017) and KITTI (Geiger, Lenz, and Urtasun 2012). Each point cloud contains 2,048 points that randomly sampled from the mesh faces and normalized into a unit sphere. We randomly generate three Euler angle rotations within $[0^\circ, 45^\circ]$ and translations within $[0, 1]$ on each axis as the rigid transformation during training. We compare our method to traditional methods: ICP (Besl and McKay 1992) and FGR (Zhou, Park, and Koltun 2016), and recent learning-based methods: PointNetLK (Aoki et al. 2019), PCRNet(Sarode et al. 2019), DeepGMR (Yuan et al. 2020), DCP (Wang and Solomon 2019), FMR (Huang, Mei, and Zhang 2020) and RPMNet (Yew and Lee 2020). In particular, we additionally included two baselines, RORNet (Wu et al. 2023b) and REGTR (Yew and Lee 2022), in the experiments on partially overlapping and real-world scenarios, respectively. These baselines encompass correspondence-based and correspondence-free scenarios, and correspondence-based methods exhibit larger network scales than correspondence-free methods. Additionally, some baselines iteratively execute the model to refine transformation precision by updating source point cloud, significantly impacting their efficiency. Therefore, these baselines contribute to demonstrating the plug-and-play nature and higher efficiency of TDM. The following evaluation metrics are used for fair comparison with previous work: Root Mean Squared Error (RMSE), Mean

Method	RMSE(**R**)	RMSE(**t**)	MAE(**R**)	MAE(**t**)	ERROR(**R**)	ERROR(**t**)
(a) Completely Overlapping Scenarios						
ICP (Besl and McKay 1992)	21.2084	0.2874	11.7468	0.1686	23.1548	0.3497
PointNetLK (Aoki et al. 2019)	7.4796	0.5820	1.7362	0.4907	4.0421	0.9741
DeepGMR (Yuan et al. 2020)	34.1993	0.4495	23.5162	0.3291	45.7786	0.6758
FGR (Zhou, Park, and Koltun 2016)	9.5964	0.1186	1.9971	0.0268	4.3337	0.0582
DCP (Wang and Solomon 2019)	15.7183	0.2002	10.7846	0.1340	20.8744	0.2740
FMR (Huang, Mei, and Zhang 2020)	6.8525	0.5851	1.5962	0.4886	3.8090	0.9751
PCRNet (Sarode et al. 2019)	19.8457	0.2288	10.2498	0.1168	19.0487	0.2442
TDM-PCRNet	3.3411	0.0459	1.6935	0.0217	3.2637	0.0451
Improvement ↑	*16.5046*	*0.1829*	*8.5563*	*0.0951*	*15.7850*	*0.1991*
RPMNet (Yew and Lee 2020)	8.0357	0.0841	1.4399	0.0170	2.9619	0.0357
TDM-RPMNet	**1.1064**	**0.0188**	**0.7211**	**0.0126**	**1.3023**	**0.0261**
Improvement ↑	*6.9293*	*0.0653*	*0.7188*	*0.0044*	*1.6596*	*0.0096*
(b) Partially Overlapping Scenarios						
ICP (Besl and McKay 1992)	24.8634	0.3520	14.8698	0.2211	29.1252	0.4610
PointNetLK (Aoki et al. 2019)	22.1867	0.5812	13.1504	0.4786	26.1196	0.9633
PCRNet (Sarode et al. 2019)	21.7402	0.2902	14.2114	0.1990	28.2444	0.4061
DeepGMR (Yuan et al. 2020)	34.6393	0.4706	24.7054	0.3465	47.8755	0.7078
FGR (Zhou, Park, and Koltun 2016)	22.7562	0.2641	8.9966	0.1110	17.2071	0.2289
DCP (Wang and Solomon 2019)	24.0230	0.3686	17.8908	0.2772	34.1244	0.5633
RORNet (Wu et al. 2023b)	4.5100	0.1613	3.2594	0.1313	6.1898	0.2571
FMR (Huang, Mei, and Zhang 2020)	15.5280	0.5772	9.5682	0.4777	19.0777	0.9595
RPMNet (Yew and Lee 2020)	11.0629	0.1912	5.7625	0.1229	11.7768	0.2598
TDM-RPMNet	**3.8624**	**0.0864**	**2.6531**	**0.0610**	**4.8977**	**0.1270**
Improvement ↑	*7.2005*	*0.1048*	*3.1094*	*0.0619*	*6.9791*	*0.1328*

Table 1: Evaluation results on ModelNet40. Bold indicates the best performance.

Absolute Error (MAE), and Relative Error/Isotropic Error (ERROR).

Evaluation on ModelNet40

TDM for Completely Overlapping Scenarios. We initially investigate the performance of TDM in the completely overlapping scenarios on ModelNet40. We opt to employ PCRNet and RPMNet, which demonstrate promising performance, to establish our denoising network and generate their corresponding diffusion variants: TDM-PCRNet and TDM-RPMNet. Table 1(a) shows quantitative results of the various algorithms. We observe TDM-RPMNet achieves the highest precision across all rotation and translation metrics. Moreover, both TDM-PCRNet and TDM-RPMNet outperform their respective baselines, PCRNet and RPMNet, by a substantial margin, particularly impressive for the 83%↑ and 86%↑ at RMSE(**R**) improvements achieved by TDM-PCRNet and TDM-RPMNet. It is worth noting that the methods augmented with TDM can achieve better performance without updating the source point cloud strategy, whereas the original method requires multiple updates to the source point cloud to refine the predicted transformations.

TDM for Partially Overlapping Scenarios. We attempt to validate the performance of TDM in currently more prevalent partially overlapping scenarios. Acquisition of partial point clouds is a common situation in the real-world, therefore this is a more extensive and difficult task. We crop the source point cloud $\mathcal{P}$ and the template point cloud $\mathcal{Q}$ respectively, and retain 70% of the points. Noted that, the quantity of points in two point clouds is still consistent after cropping. In this experiment, we exclusively employed TDM-RPMNet, as specialized methods like PCRNet designed specifically for completely overlapping scenarios lack representativeness in partially overlapping scenarios. We verify quantitatively in Table 1(b) and experimental results illustrate that TDM-RPMNet consistently outperforms the compared methods across nearly all rotation and translation metrics, yielding remarkable improvements over its baseline RPMNet.

Evaluation on 3DMatch and KITTI

We conduct experiments on the real-world datasets: 3DMatch (indoor) (Zeng et al. 2017) and KITTI odometry (outdoor) (Geiger, Lenz, and Urtasun 2012).

To be consistent with the input of ModelNet40, each input point cloud is randomly sampled to an average of 2,048 points. As shown in Table 2, we observe that TDM can still achieves significantly improved performance on real-world datasets across all rotation and translation metrics, confirming the exceptional generalization capability of our denoising network.

Ablation Studies and Disscussion

Precision vs. Efficiency. We test the efficiency of TDM in Table 3. The run time is evaluated on a NVIDIA RTX 3090 GPU with a mini-batch size of 1. The experimental results show that TDM can significantly enhance the efficiency of the original model without sacrificing precision. Such superior performance and efficiency can be primarily attributed to

Method	KITTI						3DMatch					
	RMSE (**R**)	RMSE (**t**)	MAE (**R**)	MAE (**t**)	ERROR (**R**)	ERROR (**t**)	RMSE (**R**)	RMSE (**t**)	MAE (**R**)	MAE (**t**)	ERROR (**R**)	ERROR (**t**)
ICP (1992)	18.9487	0.5466	12.6160	0.3563	20.6026	0.7752	25.4376	0.8635	13.9470	0.4202	25.7505	0.8614
PointNetLK (2019)	7.0309	0.6846	0.8580	0.5257	1.4041	1.0589	2.1669	0.5779	0.3797	0.4792	0.6542	0.9653
PCRNet (2019)	33.0369	6.8900	22.3317	5.4069	39.5509	10.7703	32.5824	1.3655	25.3004	1.0562	49.4726	2.1123
DeepGMR (2020)	46.6050	0.4386	25.8622	0.2657	44.0679	0.5785	38.3840	1.2166	26.2025	0.8146	48.0517	1.6224
FGR (2016)	13.9026	1.3483	1.7946	0.1454	3.2000	0.3053	1.5866	0.0810	0.2477	0.0099	0.4764	0.0200
DCP (2019)	18.1473	10.7517	12.8834	7.4273	22.4225	16.3362	29.7272	1.2919	23.0550	0.9689	44.0671	1.9137
FMR (2020)	7.5326	0.7692	1.4277	0.5663	2.2495	1.1334	3.3293	0.5690	0.6072	0.4712	1.0811	0.9423
REGTR (2022)	0.9315	0.0867	0.5668	0.0490	1.0870	0.0994	4.5614	0.1856	3.1127	0.1305	5.9027	0.2673
RPMNet (2020)	35.9824	0.7544	19.4641	0.3966	34.4617	0.8745	21.1454	0.4963	8.3353	0.2179	15.5514	0.4395
TDM-RPMNet	**0.8222**	**0.0248**	**0.4654**	**0.0154**	**0.9652**	**0.0335**	**0.1864**	**0.0069**	**0.1300**	**0.0044**	**0.2379**	**0.0091**
Improvement ↑	*35.1602*	*0.7296*	*18.9987*	*0.3812*	*33.4965*	*0.8410*	*20.9590*	*0.4894*	*8.2053*	*0.2135*	*15.3135*	*0.4304*

Table 2: Evaluation results on KITTI and 3DMatch. Bold indicates the best performance.

three factors: (i) TDM does not necessitate refinement transformation by updating the source point cloud many times, ensuring that the encoder of the source point cloud is not repeatedly utilized, thereby improving efficiency. (ii) The diffusion and optimization objectives (Equation 6) in $\mathbb{H}$ and $\mathbb{R}^3$ are highly reliable, requiring only a small number of sampling steps to achieve the optimal value. (iii) TDM can stably replace other refinement transformation strategies, especially those that require multiple executions of the network, significantly reducing the model inference efficiency. This experiment serves as crucial evidence demonstrating that TDM can strike a balance between precision and efficiency.

Method	MAE(**R**)	MAE(**t**)	ERROR(**R**)	ERROR(**t**)	Time(s)
ICP	11.7468	0.1686	23.1548	0.3497	0.0042
PointNetLK	1.7362	0.4907	4.0421	0.9741	0.7111
DeepGMR	23.5162	0.3291	45.7786	0.6758	0.1053
FGR	1.9971	0.0268	4.3337	0.0582	0.0916
DCP	10.7846	0.1340	20.8744	0.2740	0.2819
FMR	1.5962	0.4886	3.8090	0.9751	1.0472
PCRNet	10.2498	0.1168	19.0487	0.2442	1.2318
TDM-PCRNet	**1.6935**	**0.0217**	**3.2637**	**0.0451**	$<\mathbf{10^{-7}}$
RPMNet	1.4399	0.0170	2.9619	0.0357	0.1626
TDM-RPMNet	**0.7211**	**0.0126**	**1.3023**	**0.0261**	**0.0298**

Table 3: Precision vs. Efficiency on ModelNet40.

Any Other Diffusion Objects? In our work, TDM directly predicts object transformation from a random transformation by quaternion and translation. A natural idea arises: Is it possible to employ other transformation representation as diffusion object? We conduct an experiment in TDM-PCRNet on completely overlapping ModelNet40. Specifically, we replace quaternion (in $\mathbb{H}$) and translation (in $\mathbb{R}^3$) into rotation Euler angle (in $\mathbb{R}^3$) and translation (in $\mathbb{R}^3$), the distinction lies in the method of representing rotation. The experimental results on TDM-PCRNet are illustrated in Table 4. We observe that utilizing Euler angles does not yield performance enhancements, primarily due to the significant disparity in the scale of values between Euler angles and translation vectors. This incongruence introduces singularities into the linear noise processes for diffusion model.

Method	MAE(**R**)	MAE(**t**)	ERROR(**R**)	ERROR(**t**)
$\mathbb{H}$ and $\mathbb{R}^3$	**1.6935**	**0.0217**	**3.2637**	**0.0451**
$\mathbb{R}^3$ and $\mathbb{R}^3$	13.7668	0.1748	24.0548	0.3467

Table 4: Experimental results on different diffusion objects.

How to Fuse Noisy Transformation? As mention in previous subsection, in our work, we utilize a Transformation Encoder to obtain the noisy transformation feature $\mathcal{F}_\mathcal{G}$, and it is fused with the feature of source point cloud $\mathcal{F}_\mathcal{P}$. Additionally, we have also explored two alternative approaches: (A) concatenating all features and (B) fuse $\mathcal{F}_\mathcal{G}$ with the feature of template point cloud $\mathcal{F}_\mathcal{Q}$. The experimental outcomes are illustrated in Table 5. Practice proves that the model using approach A fails to converge and exhibits extreme instability. Furthermore, the fused approach B demonstrates no significant variation in performance.

Approaches	MAE(**R**)	MAE(**t**)	ERROR(**R**)	ERROR(**t**)
(A) $\mathcal{F}_\mathcal{G} \oplus \mathcal{F}_\mathcal{P} \oplus \mathcal{F}_\mathcal{Q}$	-	-	-	-
(B) $\mathcal{F}_\mathcal{P} \oplus (\mathcal{F}_\mathcal{G} + \mathcal{F}_\mathcal{Q})$	1.7959	0.0231	3.4632	0.0478
(C) $(\mathcal{F}_\mathcal{G} + \mathcal{F}_\mathcal{P}) \oplus \mathcal{F}_\mathcal{Q}$	1.6935	0.0217	3.2637	0.0451

Table 5: Experimental results on different approaches to fuse noisy transformation. $\oplus$ represents concatenation.

Conclusion

In this paper, we aim to push forward the development of the point cloud registration pipeline further with TDM to address balancing precision and efficiency, which formulates point cloud registration as a denoising diffusion process from noisy transformation to object transformation. Our model can be a plug-and-play agent for point cloud registration, making our method applicable to different deep registration networks.

To further explore the potential of diffusion model to solve point cloud registartion tasks, several future works are beneficial. An attempt is to apply TDM to multimodel tasks, for example, utilizing the image as the condition to guide point cloud registration.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (62276200, 62036006).

References

Amit, T.; Shaharbany, T.; Nachmani, E.; and Wolf, L. 2021. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*.

Aoki, Y.; Goforth, H.; Srivatsan, R. A.; and Lucey, S. 2019. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7163–7172.

Avrahami, O.; Lischinski, D.; and Fried, O. 2022. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18208–18218.

Bai, X.; Luo, Z.; Zhou, L.; Chen, H.; Li, L.; Hu, Z.; Fu, H.; and Tai, C.-L. 2021. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15859–15869.

Besl, P. J.; and McKay, N. D. 1992. Method for registration of 3-D shapes. In *Sensor fusion IV: Control Paradigms and Data Structures*, 586–606.

Brempong, E. A.; Kornblith, S.; Chen, T.; Parmar, N.; Minderer, M.; and Norouzi, M. 2022. Denoising Pretraining for Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4175–4186.

Chen, S.; Sun, P.; Song, Y.; and Luo, P. 2023. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19830–19843.

Choy, C.; Dong, W.; and Koltun, V. 2020. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2514–2523.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794.

Fitzgibbon, A. W. 2003. Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13-14): 1145–1153.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Graikos, A.; Malkin, N.; Jojic, N.; and Samaras, D. 2022. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*.

Guo, H.; Peng, S.; Lin, H.; Wang, Q.; Zhang, G.; Bao, H.; and Zhou, X. 2022. Neural 3d scene reconstruction with the manhattan-world assumption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5511–5520.

Guo, H.; Zhu, J.; and Chen, Y. 2023. E-LOAM: LiDAR Odometry and Mapping With Expanded Local Structural Information. *IEEE Transactions on Intelligent Vehicles*, 8(2): 1911–1921.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Huang, S.; Gojcic, Z.; Usvyatsov, M.; Wieser, A.; and Schindler, K. 2021. Predator: Registration of 3d point clouds with low overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4267–4276.

Huang, X.; Mei, G.; and Zhang, J. 2020. 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*, 11366–11374.

Jiang, H.; Salzmann, M.; Dang, Z.; Xie, J.; and Yang, J. 2024. Se (3) diffusion model-based point cloud registration for robust 6d object pose estimation. *Advances in Neural Information Processing Systems*, 36.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kurobe, A.; Sekikawa, Y.; Ishikawa, K.; and Saito, H. 2020. CorsNet: 3D point cloud registration by deep neural network. *IEEE Robotics and Automation Letters*, 5(3): 3960–3966.

Luo, S.; and Hu, W. 2021. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2837–2845.

Lyu, Z.; Kong, Z.; Xu, X.; Pan, L.; and Lin, D. 2021. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*.

Mei, G. 2021. Point cloud registration with self-supervised feature learning and beam search. In *Digital Image Computing: Techniques and Applications*, 01–08. IEEE.

Mei, G.; Huang, X.; Zhang, J.; and Wu, Q. 2022. Partial Point Cloud Registration Via Soft Segmentation. In *IEEE International Conference on Image Processing*, 681–685. IEEE.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, 8162–8171. PMLR.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 652–660.

Rusinkiewicz, S. 2019. A symmetric objective function for ICP. *ACM Transactions on Graphics*, 38(4): 1–7.

Rusinkiewicz, S.; and Levoy, M. 2001. Efficient variants of the ICP algorithm. In *Proceedings of International Conference on 3-D Digital Imaging and Modeling*, 145–152.

Sarode, V.; Li, X.; Goforth, H.; Aoki, Y.; Srivatsan, R. A.; Lucey, S.; and Choset, H. 2019. Pcrnet: Point cloud registration network using pointnet encoding. *arXiv preprint arXiv:1908.07906*.

Segal, A.; Haehnel, D.; and Thrun, S. 2009. Generalized-icp. In *Robotics: Science and Systems*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Vahdat, A.; Williams, F.; Gojcic, Z.; Litany, O.; Fidler, S.; Kreis, K.; et al. 2022. LION: Latent point diffusion models for 3D shape generation. *Advances in Neural Information Processing Systems*, 35: 10021–10039.

Wang, Y.; and Solomon, J. M. 2019. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3523–3532.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S. E.; Bronstein, M. M.; and Solomon, J. M. 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions On Graphics*, 38(5): 1–12.

Wolleb, J.; Sandkühler, R.; Bieder, F.; Valmaggia, P.; and Cattin, P. C. 2022. Diffusion models for implicit image segmentation ensembles. In *International Conference on Medical Imaging with Deep Learning*, 1336–1348. PMLR.

Wu, Y.; Yuan, Y.; Fan, X.; Huang, X.; Gong, M.; and Miao, Q. 2023a. PCRDiffusion: Diffusion Probabilistic Models for Point Cloud Registration. *arXiv preprint arXiv:2312.06063*.

Wu, Y.; Zhang, Y.; Ma, W.; Gong, M.; Fan, X.; Zhang, M.; Qin, A.; and Miao, Q. 2023b. Rornet: Partial-to-partial registration network with reliable overlapping representations. *IEEE Transactions on Neural Networks and Learning Systems*.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1912–1920.

Xu, H.; Liu, S.; Wang, G.; Liu, G.; and Zeng, B. 2021. Omnet: Learning overlapping mask for partial-to-partial point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3132–3141.

Yang, J.; Li, H.; Campbell, D.; and Jia, Y. 2015. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11): 2241–2254.

Yew, Z. J.; and Lee, G. H. 2020. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11824–11833.

Yew, Z. J.; and Lee, G. H. 2022. Regtr: End-to-end point cloud correspondences with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6677–6686.

Yuan, W.; Eckart, B.; Kim, K.; Jampani, V.; Fox, D.; and Kautz, J. 2020. Deepgmr: Learning latent gaussian mixture models for registration. In *European Conference on Computer Vision*, 733–750.

Yuan, Y.; Wu, Y.; Fan, X.; Gong, M.; Ma, W.; and Miao, Q. 2023. EGST: Enhanced Geometric Structure Transformer for Point Cloud Registration. *IEEE Transactions on Visualization & Computer Graphics*, (01): 1–13.

Yuan, Y.; Wu, Y.; Fan, X.; Gong, M.; Miao, Q.; and Ma, W. 2024a. Inlier Confidence Calibration for Point Cloud Registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5312–5321.

Yuan, Y.; Wu, Y.; Gong, M.; Miao, Q.; and Qin, A. K. 2024b. One-nearest neighborhood guides inlier estimation for unsupervised point cloud registration. *IEEE Transactions on Neural Networks and Learning Systems*.

Zeng, A.; Song, S.; Nießner, M.; Fisher, M.; Xiao, J.; and Funkhouser, T. 2017. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1802–1811.

Zhou, Q.-Y.; Park, J.; and Koltun, V. 2016. Fast global registration. In *European Conference on Computer Vision*, 766–782.

Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M. R.; and Pollefeys, M. 2022. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12786–12796.