P2P-Bridge: Diffusion Bridges for 3D Point Cloud Denoising

Mathias Vogel[1], Keisuke Tateno[2], Marc Pollefeys[1,3], Federico Tombari[2,4], Marie-Julie Rakotosaona[*,2], and Francis Engelmann[*,1,2]

[1] ETH Zurich [2] Google [3] Microsoft [4] TU Munich

Abstract. In this work, we address the task of point cloud denoising using a novel framework adapting Diffusion Schrödinger bridges to unstructured data like point sets. Unlike previous works that predict point-wise displacements from point features or learned noise distributions, our method learns an optimal transport plan between paired point clouds. In experiments on object datasets such as the PU-Net dataset and real-world datasets like ScanNet++ and ARKitScenes, P2P-Bridge improves by a notable margin over existing methods. Although our method demonstrates promising results utilizing solely point coordinates, we demonstrate that incorporating additional features like RGB information and point-wise DINOV2 features further improves the results.Code and pretrained networks are available at https://github.com/matvogel/P2P-Bridge.

Keywords: Point Cloud Denoising, Diffusion Models, Schrödinger Bridges

1 Introduction

The use of point clouds to represent 3D objects and scenes [9,24,54] is widespread across various fields, including 3D vision, robotics [28, 75], augmented/virtual reality, and autonomous driving [26,59]. Recently, 3D scanning devices such as LIDAR sensors have gained popularity and have been incorporated into off-the-shelf consumer products. Using these handheld devices, users can scan objects or venues in a relatively short amount of time. However, the resulting point clouds often contain substantial noise due to hardware limitations such as low scanner resolution, sensor noise, limited range, or environmental factors, such as reflections, scattering, or occlusions which can have detrimental effects on down-stream tasks that rely on good quality point clouds [14,53,62]. To address this issue, point cloud denoising has emerged as a critical technique that reduces noise in scanned objects and enhances geometric details.

While there have been notable advancements in point cloud denoising research, cleaning up scans corrupted by real-world scanner noise remains challenging due to the need to grasp the underlying topology and nature of the underlying clean surface as well as the characteristics of the noise. Although conventional

[*] Equal contribution.

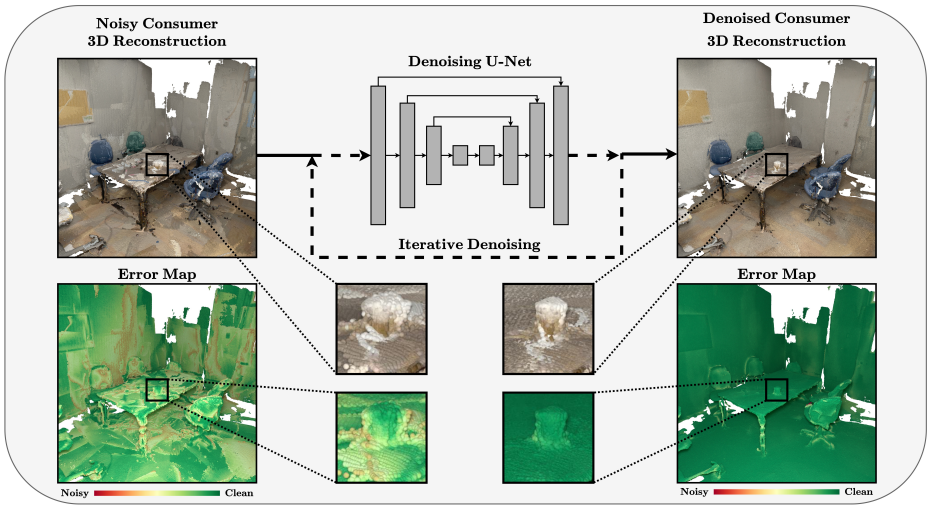

Fig. 1: Illustration of P2P-Bridge applied to a noisy LIDAR scan.

point cloud denoising methods [2, 10, 17, 18, 31, 57, 66, 70, 71] may perform well in specific circumstances, they often require extensive fine-tuning of parameters or additional point features such as normals and often fail to generalize to complex noise patterns.

Deep learning approaches [34, 36, 38, 45, 68] have shown superior performance over traditional methods due to data-driven approach. One class of deep learning approaches [19, 34] tackles denoising by first resampling the point cloud to a coarse set of points, potentially eliminating high-frequency noise. They recover the underlying clean surface by upsampling and refining the point cloud. Other methods [13, 45, 66] try to recover clean data by regression or point-wise displacement prediction, whereas PointCleanNet [45] incorporates outlier removal too. Recently, score-based [36, 68] and flow-based [38] models have shown exciting results by learning the score or probability of the noise distribution directly. However, current models that address noise in point clouds are trained under the assumption of synthetic noise, such as isotropic Gaussian noise. Our experiments reveal that this assumption is often insufficient for denoising real-world 3D scans obtained from off-the-shelf devices, as it neglects effects such as clusters of outliers, ghost points, or edge flares [2]. Furthermore, previous methods are often trained on minimizing distance metrics that scale worse than linearly with the size of input [46] inhibiting scaling the model architecture, which plays a crucial role in point-feature learning [44]. Finally, recent models often focus on learning denoising tasks from point-based features like colors or normals only, which do not account for the high-level semantic properties of the underlying data [41, 61]. We propose to exploit high-level learned features by incorporating point-wise features extracted from DINOV2 [41].

This work proposes a novel supervised approach for point cloud denoising based on diffusion models [8, 20]. We approach the denoising task by formulating it as a Schrödinger bridge problem [8, 48] and solve it by training a network to find an optimum transport plan between the noisy and corresponding clean point cloud, allowing our method to be trained on any data, such as indoor scenes (*c.f.* Fig. 1). We incorporate RGB and DINOV2 [41] features to improve our method further. Experiments show that our approach outperforms other state-of-the-art in synthetic and real-world scenarios.

In summary, our main contributions are:

1. We propose P2P-Bridge (Pointcloud-to-Pointcloud Bridge), a new approach for point cloud denoising inspired by the Schrödinger bridge problem formulating point cloud denoising as a tractable data-to-data diffusion process.
2. Furthermore, we advocate for semantically informed denoising by incorporating high-level features, such as DINOV2, to guide the denoising process.

2 Related Works

Traditional denoising methods can be roughly categorized into filter-based and optimization-based approaches. The filter-based methods draw from image and signal processing, assuming that the clean point cloud is corrupted with high-frequency noise. Bilateral filter approaches [2, 18, 67] are suitable for denoising object surfaces while preserving sharp edges. Guided filter-based approaches [17, 31, 57, 70, 71] attempt to fit a local linear model to a noisy point cloud aiming to preserve local details by using guidance from point coordinates or normals. Graph-based methods [12, 23, 64] model the point cloud as a graph to capture the underlying geometric structure and the relation of points with each other. Optimization-based methods range from sparse reconstruction [11, 27, 39] to non-local-based point cloud denoising methods [3, 33, 64]. However, all these traditional methods rely on manually tuned hyper-parameters, which are tedious to obtain and typically do not generalize well.

Deep-learning-based methods have recently shown promising results and improvement over traditional denoising methods. PointCleanNet [45] first removes outliers and then predict point-wise displacement vectors to denoise the point clouds. TotalDenoising [19] is an unsupervised method that applies total least squares [16] to unstructured data such as point clouds. DMRDenoise [34] uses downsampling by differential pooling to estimate a manifold from which it resamples points to obtain a denoised point cloud. Score-matching-based methods [52] learn the score function of a tractable noise distribution and use (momentum) gradient ascent [36, 68] to predict local displacements during inference. PD-Flow [38] utilizes normalizing flows to estimate the noise probability density function directly by disentangling the noise from the clean point cloud in a latent space. I-PFN [49] improves upon iterative denoising methods using separated iteration modules for each denoising iteration already during training.

All mentioned methods except PD-Flow are all trained under the assumption of Gaussian noise since it is easy to generate training pairs of clean and noisy point

clouds. However, as we will show experimentally, this does not necessarily translate to complex noise in real-world indoor scenes. The main difference between our method and most previous works is that our method can be applied to any general data-to-data problem. By learning data-specific noise characteristics, our method better recovers the underlying clean data, removing outlier clusters and recovering fine details. Lastly, our method performs denoising using DDPM sampling [20], making it more robust to the number of denoising steps as ablation studies show. Our method shows good results with as few as three function evaluations.

3D reconstruction involves creating a three-dimensional representation of real-world scenes using 2D images and additional data such as depth. It differs from 3D point cloud denoising, but can serve as the initial step in generating 3D point clouds from real-world scenes. As a result, we will be discussing some relevant works in this field. 3DMatch [63] is a data-driven approach for matching RGB-D reconstructions using learned volumetric features. RoutedFusion [56] and Map-Adapt [69] introduce machine learning-based approaches for real-time depth map fusion, and uses a neural network to predict non-linear updates for voxel-based fusion, addressing common errors and artifacts, especially for thin objects and edges. NICE-SLAM [74] is a hierarchical grid-based SLAM method that uses RGB-D data for accurate environment reconstruction. It incorporates pre-trained models to enhance spatial understanding, improving mapping and tracking efficiency.

NICER-SLAM [73] uses RGB input data to optimize an end-to-end joint mapping and tracking system, enabling it to predict colors, depths, and normals. Additional losses, such as warping and optical flow loss, further enhance the geometric consistency of NICER-SLAM.

3 Method

3.1 Overview

Let the distributions of noisy point sets $\tilde{\mathcal{P}} = \{\tilde{x}_i\} \in \mathbb{R}^{N \times D}$ and clean point sets $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M \times D}$, where D is the point feature dimension and N and M are the number of points in the noisy and clean point cloud respectively. We aim to denoise the noisy point sets $\tilde{\mathcal{P}}$ by using diffusion models [20,35,51] and exploiting their nature of predicting clean data from noisy priors. Diffusion models are successfully used in many image generation and translation tasks [30, 40, 47] and recently also in point cloud generation and completion [25, 37, 55, 65, 65, 72]. While most diffusion-based methods, as well as related works on point cloud denoising [36, 45, 68] use Gaussian priors, we argue that employing a data-to-data instead of a data-to-noise approach is more suited for point cloud denoising, especially when dealing with specific sensor data. Using the distribution of noisy point sets $\tilde{\mathcal{P}}$ as prior distribution enables our method to learn data-dependent real-world noise characteristics. However, for training a diffusion model, the process of diffusing a clean sample to a noisy sample generally has to be tractable, as diffusion models are trained to learn step-wise noise removal. For real-world

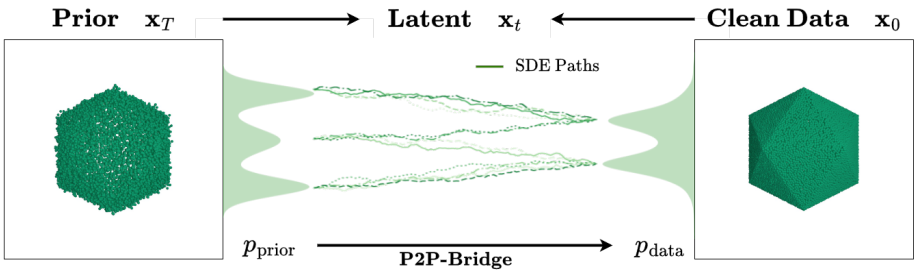

Fig. 2: Illustration of P2P-Bridge, modeling point cloud denoising as a reverse data-to-data diffusion process. Our model can effectively transform noisy data into cleaner data by learning a bridge between clean and noisy data.

data, regardless, this process is unknown. One method to simulate a diffusion process between clean and noisy samples is using a Schrödinger bridge (SB). Schrödinger bridges have been increasingly used in generative models for image-to-image translation [4,8,30], protein matching [50], or recently text-to-speech [6]. To our knowledge, we are the first to use this approach for point cloud denoising.

3.2 Pointcloud-to-Pointcloud Bridges

Tractable diffusion bridges. We consider the point cloud denoising problem from the perspective of diffusion models [20]. By generating a diffusion process over T timesteps $\{\mathbf{x}_1, \dots, \mathbf{x}_T\}$ with $\mathbf{x}_t \in \mathbb{R}^{N\times 3}$, a sample from clean data $\mathbf{x}_0 \sim p_{\text{data}}$ gets diffused into a noisy sample $\mathbf{x}_T \sim p_{\text{prior}}$. In the case of point cloud denoising, the prior distribution corresponds to the distribution over noisy point sets $\tilde{\mathcal{P}}$ (*c.f.* Fig. 2). Considering a reference path measure $p_{\text{ref}}(\mathbf{x}_{0:T})$ which describes this process, our goal is to find a process $p^*(\mathbf{x}_{0:T})$ such that $p^*(\mathbf{x}_0) = p_{\text{data}}$ and $p^*(\mathbf{x}_T) = p_{\text{prior}}$ minimizing the Kullback-Leibler divergence between p_{ref} and p^*. This problem is also known as Schrödinger's bridge (SB) problem [29,48]. It can be described using the forward and backward stochastic differential equations (SDEs) defined as

$$\begin{aligned} \mathrm{d}\mathbf{x}_t &= [\mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t + g^2(t)\nabla \log \Psi_t(\mathbf{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{data}} \\ \mathrm{d}\mathbf{x}_t &= [\mathbf{f}(\mathbf{x}_t, t)\mathrm{d}t - g^2(t)\nabla \log \hat{\Psi}_t(\mathbf{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t, \quad \mathbf{x}_t \sim p_{\text{prior}} \end{aligned} \tag{1}$$

where $\mathbf{w_t}$ is a Wiener process, $\mathbf{f}$ is a vector-valued function called *drift* and $\mathbf{g}$ is a scalar-valued term known as the *diffusion* coefficient. The terms $\nabla \log \Psi_t(\mathbf{x}_t)$ and $\nabla \log \hat{\Psi}_t(\mathbf{x}_t)$ are additional nonlinear drift terms that solve the following coupled partial differential equations (PDEs)

$$\begin{cases} \frac{\delta \Psi}{\delta t} &= -\nabla_x \Psi^{\mathrm{T}} \mathbf{f} - \frac{1}{2}\mathrm{Tr}(g^2 \nabla_x^2 \Psi) \\ \frac{\delta \hat{\Psi}}{\delta t} &= -\nabla_x \hat{\Psi}^{\mathrm{T}} \mathbf{f} + \frac{1}{2}\mathrm{Tr}(g^2 \nabla_x^2 \hat{\Psi}) \end{cases} \tag{2}$$

such that $\Psi_0\hat{\Psi}_0 = p_{\text{data}}, \Psi_T\hat{\Psi}_T = p_{\text{prior}}$ and $p_t = \Psi_t\hat{\Psi}_t$. Chen *et al.* [4] show that Eq. (1) is a generalization of score-based generative modeling (SGM) [51] to nonlinear processes. Directly solving the differential equation system is not practicable and computationally expensive. However, recent works [6,30] introduce simplified tractable frameworks under the assumption that we deal with paired boundary data *i.e.* $p(\mathbf{x}_0, \mathbf{x}_T) = p_{\text{data}}(\mathbf{x}_0)p_{\text{prior}}(\mathbf{x}_T \mid \mathbf{x}_0)$. In the context of point clouds, this means that the distribution over noisy point clouds is modeled as a joint distribution of clean point sets ($p_{\text{data}}(\mathbf{x}_0)$) and noise ($p_{\text{prior}}(\mathbf{x}_T \mid \mathbf{x}_0)$). When the boundary data is as a mixture of Diracs ($\delta_{\mathbf{x}_0}, \delta_{\mathbf{x}_T}$) with $\mathbf{f} := 0$ and using a linear diffusion schedule $g^2(t)$, it can be shown [6] that the posterior of Eq. (1) has an analytic form given by

$$q(\mathbf{x}_t \mid \mathbf{x}_0, x_T) = \mathcal{N}(\mathbf{x}_t; \mu_t(\mathbf{x}_0, \mathbf{x}_T), \Sigma_t) \tag{3}$$

with

$$\mu_t = \frac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + \sigma_t^2}\mathbf{x}_0 + \frac{\sigma_t^2}{\bar{\sigma}_t^2 + \sigma_t^2}x_T \quad \text{and} \quad \Sigma_t = \frac{\sigma_t^2\bar{\sigma}_t^2}{\sigma_t^2 + \bar{\sigma}_t^2} \tag{4}$$

where $\sigma_t^2 = \int_0^t g^2(\tau)\mathrm{d}\tau$ and $\bar{\sigma}_t^2 = \int_t^1 g^2(\tau)\mathrm{d}\tau$. This simplifies Eq. (2) and makes it fully tractable. We can parameterize a network ϵ_θ that predicts the noise added to $\mathbf{x}_0$ at timestep t resulting in the noisy sample $\mathbf{x}_t$ using the noise-prediction loss:

$$\mathcal{L} = \|\epsilon_\theta(\mathbf{x}_t, t) - \frac{\mathbf{x}_t - \mathbf{x}_0}{\sigma_t}\|_2^2. \tag{5}$$

During inference, we can iteratively sample using DDPM sampling [20]

$$p(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \hat{\mathbf{x}}_0) = \mathcal{N}(\mathbf{x}_t; \mu_t(\hat{\mathbf{x}}_0, \mathbf{x}_T), \Sigma_t), \quad \hat{\mathbf{x}}_0 = \mathbf{x}_t - \sigma_t\epsilon_\theta(\mathbf{x}_t, t), \tag{6}$$

as this induces the same marginal density of SB paths as long as $\hat{\mathbf{x}}_0$ is close to the actual $\mathbf{x}_0$ [6,30]. However, sampling from μ_t describes an interpolation between two point clouds. This operation is straightforward for data with a fixed domain, such as image data, where pixels are attached to a static grid. It is, however, not well defined for unordered point clouds and depends on a proper distance metric [5].

Meaningful interpolation between unordered point sets. We use the shortest path interpolation method from PointMixup [5] to describe the path of the posterior mean μ_t. Shortest path interpolation tries to find an optimum assignment ϕ^* that minimizes the average distance for each point in the point cloud $\mathbf{x}_T$ to its nearest neighbor in $\mathbf{x}_0$. Assuming that the noisy and clean point sets both contain N points, the assignment problem is defined as

$$\phi^* = \arg\min_{\phi \in \mathbf{\Phi}} \sum_{i=1}^{N} \|\mathbf{x}_T^i - \mathbf{x}_0^{\phi(i)}\|_2, \tag{7}$$

where $\mathbf{\Phi} = \{\{1, \ldots, N\} \to \{1, \ldots, N\}\}$ is the set of possible bijective assignments between points in $\mathbf{x}_T$ and $\mathbf{x}_0$. Using shortest-path-interpolation resembles finding an optimal transport plan between two point sets when the cost is a squared

geodesic distance [42]. The resulting path from shortest-path-interpolation corresponds to the path taken by the posterior of Eq. (1) when the stochasticity of the bridge vanishes *i.e.* $g^2(t) \to 0$ [6, 30, 42] which motivates the choice of nearest-path-interpolation over other possible interpolation methods. Diminishing the stochasticity of the bridge effectively reduces the bridge SDE to an optimal transport ordinary differential equation (OT-ODE) of the form

$$\mathrm{d}\mathbf{x}_t = \frac{g^2(t)}{\sigma_t^2}(\mathbf{x}_t - \mathbf{x}_0)\mathrm{d}t. \tag{8}$$

In practice, we have to calculate the optimal assignment for every data pair in our dataset only once. Subsequently, we can employ ϕ^* to reorder the clean point clouds so that they are aligned with their corresponding noisy point clouds. During training, we can sample $\mathbf{x}_t$ without solving the optimum assignment problem again, allowing fast and scalable training. Further discussion and experiments on shortest-path interpolation can be found in the appendix.

3.3 Implementation

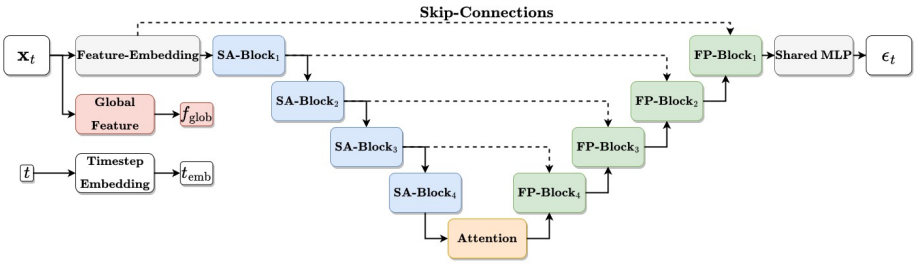

Fig. 3: The network architecture, based on PointVoxelConvolutions (PVC) [32]. We adapt the network implementation from LION [65], augmenting it with multi-headed global attention and a feature embedding module. Both feature embedding and the final shared MLP block are implemented using 1×1 convolutions.

Model architecture. We follow previous works on point cloud diffusion models [37,65,72] and use a model architecture (*c.f.* Fig. 3) based on the PointVoxel-CNN (PVCNN) [32]. PVCNN is a PointNet++ [43] inspired architecture that augments the set abstraction (SA) and feature propagation (FP) blocks with global features extracted from a vocalized point cloud representation. Similar to LION [65], we incorporate squeeze-excitation blocks [22] and a global feature extraction network. In addition, we employ a feature embedding layer mapping the incoming features to a higher dimension using a 1×1 convolution. We follow recent works on image diffusion models [21] and only utilize attention on the lowest layer employing multiple attention heads. The network is conditioned on timestep t

using sinusoidal positional embeddings and point features. Global features are incorporated via adaptive group normalization. Input data is processed in patches, sampling points from dataset-dependent radius spheres to ensure correspondence between noisy and clean data in space.

Input features. Additional RGB data is often available from mobile phone scans. We propose point-wise features extracted by DINOV2 [41] from the raw RGB features. The pixel-wise DINOV2 features are projected to the noisy point cloud using camera poses and intrinsics, resulting in point-wise features.

4 Experiments

4.1 Datasets

For evaluation, we compare our P2P-Bridge on well-established 3D object datasets depicting single objects and indoor scene datasets. For object-level denoising, we use the PU-Net dataset [60], which consists of 40 objects for training and 20 objects for evaluation. We use the PC-Net dataset [45] to provide another ten objects for model evaluation only. For both object datasets, we follow the commonly used practice and simulate noise using isotropic Gaussians.

Unlike prior work, we propose to evaluate on scene-level point clouds additionally. This setup is closer to real-world usage scenarios. For evaluation on real-world datasets, we choose the indoor-scene datasets ScanNet++ [58] and ARKitScenes [1] because they provide paired clean and noisy point cloud data as well as pose data. ScanNet++ contains 330 indoor scenes, where each scene has a series of noisy depth maps obtained by a handheld LIDAR scanner and a clean scan obtained by a Faro laser scanner. We use the 3D reconstruction script provided by the authors of ScanNet++ to construct the noisy point clouds, on which we then apply the denoising methods. Additionally, we evaluate when the reconstruction is refined using 3DMatch [63]. ARKitScenes contains 5047 scans of different indoor venues, where the noisy scans are obtained using Apple ARKit surface reconstruction. The clean scan is acquired by a Faro laser scanner. Further details can be found in the supplementary materials.

4.2 Evaluation Metrics

We use the Chamfer distance (CD) and the Point-to-Mesh (P2M) distance as quantitative evaluation metrics. The CD measures the similarity between a predicted point cloud $\hat{\mathcal{P}} = \{\hat{x}_i \in \mathbb{R}^3\}_{i=1}^n$ and a clean point cloud $\mathcal{P} = \{x_j \in \mathbb{R}^3\}_{j=1}^m$ as

$$\mathrm{CD}(\hat{\mathcal{P}}, \mathcal{P}) = \frac{1}{2n} \sum_{i=1}^{n} \|\hat{x}_i - \mathrm{NN}(\hat{x}_i, \mathcal{P})\|_2^2 + \frac{1}{2m} \sum_{j=1}^{m} \|x_j - \mathrm{NN}(x_j, \hat{\mathcal{P}})\|_2^2, \tag{9}$$

where NN is the nearest-neighbor function. The first term ($\overrightarrow{\mathrm{CD}}$) approximately describes the average distance of noisy points to the ground truth surface, and

the second term ($\overleftarrow{\text{CD}}$) encourages uniform covering. The Point-to-Mesh (P2M) distance is defined as

$$\text{P2M}(\hat{\mathcal{P}}, \mathcal{M}) = \frac{1}{2n} \sum_{i=1}^{n} \min_{f \in \mathcal{M}} d(\hat{x}_i, f) + \frac{1}{2|\mathcal{M}|} \sum_{f \in \mathcal{M}} \min_{\hat{x}_i \in \hat{\mathcal{P}}} d(\hat{x}_i, f) \tag{10}$$

where $d(x, f)$ is a function measuring the distance of point x to face f. The first term, therefore, describes Face-to-Point distance (F2P), and the second term corresponds to the Point-to-Face distance (P2F). For calculating object-level metrics, we center and scale the prediction and ground truth to the unit sphere following ScoreDenoise [36].

4.3 Experimental Details

We train our model on the PU-Net dataset to denoise artificially noised objects with Gaussian noise, following previous works. For ScanNet++ and ARKitScenes, we train all deep-learning-based denoising methods, including ours, with a batch size of 32 for a maximum of 100,000 steps. We use the training parameters and model weights provided in the publicly available code bases for previous works [34, 36, 38, 68]. Additional experimental details are provided in the supplementary materials.

4.4 Comparison on Objects

We quantitatively evaluate our method with traditional methods such as Bilateral [10] or GLR [64] as well as deep-learning-based methods including PC-Net [45], DMR [34], ScoreDenoise [36], MAG [68] and PD-Flow [38]. For the evaluation, we choose Gaussian noise levels ranging from 1% to 3% of the object's bounding box diagonal for sparse (10k points) and dense (10k points) objects. Table 1 shows that our method outperforms previous optimization-based methods and deep-learning-based methods in most noise settings. In the 1% setting, our method performs second best to PD-Flow, whereas, for higher noise levels, we see a significant increase in measured accuracy compared to previous ones. Our method also seems to adapt better to unseen objects, as indicated by the results on the PC-Net dataset. Note that those results are achieved with only three denoising steps. Figure 4 qualitatively compares our method and most recent deep-learning-based methods on denoising objects corrupted by 3% of isotropic Gaussian noise. Our method seems to generate less noisy and smoother results than previous works. Additional qualitative results and experiments using different noise types as well as run-times are provided in the supplementary materials.

4.5 Comparison on Indoor Scenes

To further investigate our method's denoising capabilities and that of previous works, we evaluate the methods when applied to reconstructions of large-scale indoor scenes. This setting comes with additional challenges and noise sources

Table 1: Object-level Scores. We show the Chamfer distance (CD) and Point-2-Mesh distance (P2M) on the PU-Net *(top)* and PC-Net *(bottom)* datasets. Scores are multiplied by 10^4. When possible, baseline scores are taken from [36] and [68], otherwise, we use the publicly available weights and testing scripts to evaluate on the test data provided by [36].

	Num. of Points	$10 \cdot 10^3$ (sparse)						$50 \cdot 10^3$ (dense)					
	Gaussian Noise	1%		2%		3%		1%		2%		3%	
	Method	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M	CD	P2M
PU-Net [60]	Bilateral [10]	3.65	1.34	5.01	2.02	7.00	3.56	0.88	0.23	2.38	1.39	6.30	4.73
	PCNet [45]	3.52	1.15	7.47	3.97	13.1	8.74	1.05	0.35	1.45	0.61	2.29	1.29
	DMRDenoise [34]	4.48	1.72	4.98	2.12	5.89	2.85	1.16	0.47	1.57	0.80	2.43	1.53
	GLR [64]	2.96	1.05	3.77	1.31	4.91	2.11	0.70	0.16	1.59	0.83	3.84	2.71
	ScoreDenoise [36]	2.52	0.46	3.69	1.07	4.71	1.94	0.72	0.15	1.29	0.57	1.93	1.04
	MAG [68]	2.50	0.46	3.63	1.05	4.69	1.92	0.71	0.15	1.29	0.56	1.93	1.05
	PD-Flow [38]	**2.13**	0.38	3.25	1.01	5.19	2.52	0.65	0.16	1.42	0.78	3.90	2.86
	I-PFN [49]	2.31	**0.37**	3.43	0.9	5.49	2.5	0.66	0.12	1.05	0.43	2.54	1.65
	P2P-Bridge (Ours)	**2.28**	**0.39**	**3.20**	**0.81**	**3.99**	**1.42**	**0.59**	**0.09**	**0.90**	**0.32**	**1.56**	**0.84**
PC-Net [45]	Bilateral [10]	4.32	1.35	6.17	1.65	8.30	2.39	1.17	0.20	2.50	0.63	6.08	2.19
	PCNet [45]	3.85	1.22	8.75	3.04	14.5	5.87	1.29	0.29	1.91	0.51	3.25	1.08
	DMRDenoise [34]	6.60	2.15	7.15	2.24	8.09	2.49	1.57	0.35	2.01	0.49	2.99	0.86
	GLR [64]	3.40	0.96	5.27	1.15	7.25	1.67	0.96	0.13	2.02	0.42	4.50	1.31
	ScoreDenoise [36]	3.37	0.83	5.13	1.20	6.78	1.94	1.07	0.18	1.66	0.35	2.49	0.66
	MAG [68]	3.37	0.83	5.13	1.19	7.24	1.94	1.07	0.18	1.66	0.35	3.56	1.15
	PD-Flow [38]	3.24	**0.62**	4.62	0.92	6.61	1.62	0.97	0.15	1.80	0.40	4.28	1.37
	I-PFN [49]	3.05	0.72	4.95	1.16	7.39	2.21	0.99	0.14	1.43	0.27	3.03	0.86
	P2P-Bridge (Ours)	**2.88**	**0.63**	**4.47**	**0.89**	**5.58**	**1.29**	**0.92**	**0.12**	**1.35**	**0.24**	**2.12**	**0.49**

like clusters of outliers or surface thickening effects [2] and shows the ability of the methods to scale to large inputs.

On ScanNet++, we evaluate all models on the noisy point cloud reconstructions provided by the authors. These reconstructions were obtained by filtering the depth maps according to their agreement with the Faro laser depths, followed by projection using globally optimized poses without applying further fusion methods [58]. Additionally, we evaluate all methods on reconstructions obtained by applying 3DMatch [63] on the pre-filtered depth maps, using the globally optimized poses. On ARKitScenes, we directly apply the methods to the ARKit reconstructions provided by the authors as part of the 3D object detection subset. For further details about these reconstructions, we refer to the corresponding paper [1]. Table 2 shows that our method using RGB+DINO features mostly achieves the best results, followed by our method only using RGB or coordinate features. We qualitatively compare the best-performing methods in Fig. 5, including the noisy and Faro point clouds. We use a color gradient to represent the distance between the predicted and the ground truth points, ranging from green to red, for low and high distances, respectively. ScoreDenoise, as well as all other methods that are trained under the assumption of Gaussian noise, suffer from pattern-like artifacts. Due to memory constraints, all deep-learning methods denoise large point clouds in patches. For methods trained under synthetic noise,

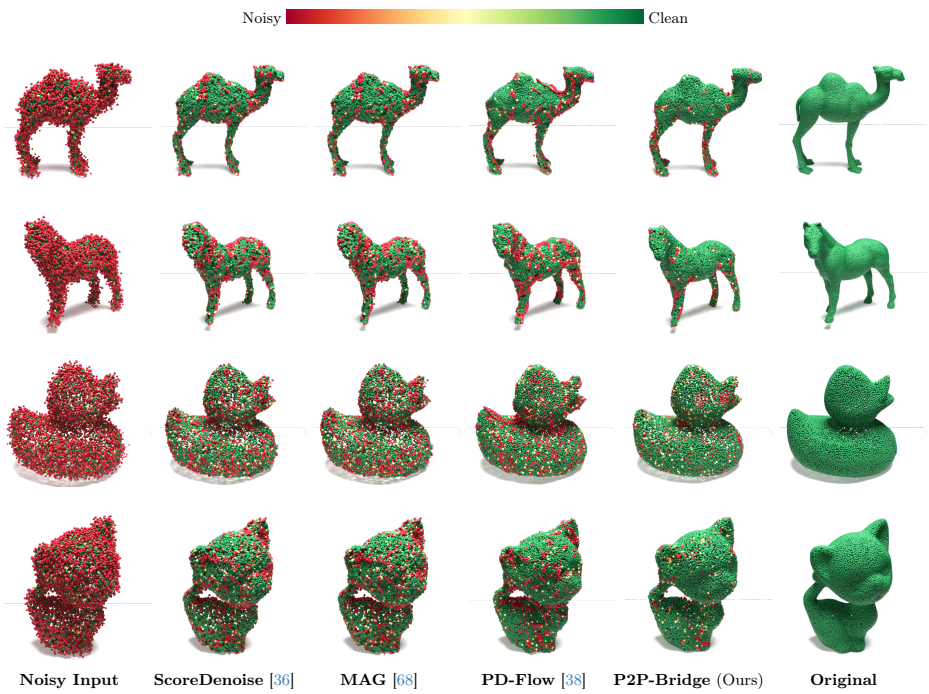

Fig. 4: Qualitative comparison of our P2P-Bridge and recent deep-learning-based point cloud denoising methods on the PU-Net dataset under 3% isotropic Gaussian noise.

this leads to a situation where points at the border of each patch are assumed to be outliers and concentrate on artificial borders around these patches. This effect is further enhanced when Langevin sampling without stochasticity is used as in ScoreDenoise and MAG, leading to a collapse of points [52]. We hypothesize that our method is less susceptible to patch artifacts for two reasons. First, we do not train under the assumption of Gaussian noise, making our method more robust in differentiating real object borders and borders that arise due to patch-based processing. Furthermore, instead of simply accumulating predictions over patches, followed by farthest-point sampling, we average the predicted point coordinates for every point in the noisy cloud. Although PD-Flow exhibits circular patch artifacts, it does not suffer from points collapsing on patch borders due to training on real-world noise. Due to the lower amount of detail in the noisy scans of ARKitScenes (*c.f.* Fig. 5) compared to ScanNet++, the denoising is generally less pronounced. Nonetheless, our method produces sharper edges of objects and smoother surfaces compared to the other methods. There are also incomplete objects visible, which none of the methods can complete. To tackle this, future works could incorporate strategies from point cloud completion [15, 37] to further improve the results.

Table 2: Indoor Scenes Scores. Quantitative point cloud denoising comparisons on scenes from the ScanNet++ [58] and ARKitScenes [1] test set. The Faro scanner point clouds act as a reference for the ground truth. $\overrightarrow{\text{CD}}$ describes the average distance of noisy points to the ground truth surface and $\overleftarrow{\text{CD}}$ describes the average distance from ground truth points to noisy points. Similarly, P2F is the Point-to-Face distance and F2P is the Face-to-Point distance. Metrics on ScanNet++ are multiplied by 10^4, on ARKitScenes the factor is 10^3.

	Dataset	ScanNet++ [58] Apple LiDAR						ScanNet++ [58] Apple LiDAR + 3DMatch [63]						ARKitScenes [1] Apple LiDAR		
	Input Source															
Method	Features	P2F	F2P	$\overrightarrow{\text{CD}}$	$\overleftarrow{\text{CD}}$	CD	P2M	P2F	F2P	$\overrightarrow{\text{CD}}$	$\overleftarrow{\text{CD}}$	CD	P2M	$\overrightarrow{\text{CD}}$	$\overleftarrow{\text{CD}}$	CD
Bilateral [10]	XYZ	6.29	140.59	6.66	145.44	73.44	76.05	108.70	**18.32**	108.89	**19.67**	64.28	63.51	15.87	70.49	43.18
DMR [34]	XYZ	6.48	149.99	6.71	159.13	78.24	82.92	99.96	19.61	100.16	21.27	60.71	59.79	10.84	30.51	20.68
ScoreDenoise [36]	XYZ	3.49	128.59	3.72	132.71	68.21	66.04	97.11	18.87	97.31	20.26	58.78	57.99	9.56	30.86	20.21
MAG [68]	XYZ	5.43	147.54	5.66	152.07	78.87	76.49	99.05	24.82	99.26	26.69	62.97	61.93	9.57	30.82	20.20
PD-Flow [38]	XYZ	3.80	147.49	4.02	151.90	77.96	75.64	85.29	21.00	85.49	22.56	54.02	53.14	9.93	33.82	21.87
I-PFN [49]	XYZ	3.80	132.98	4.03	137.21	70.62	68.39	83.99	18.99	84.19	20.43	52.31	51.49	**9.19**	31.99	20.59
P2P-Bridge (Ours)	XYZ	2.48	**122.23**	2.71	**126.22**	64.46	62.35	50.87	18.69	51.07	20.05	35.56	34.78	9.65	30.64	20.14
P2P-Bridge (Ours)	XYZ, RGB	2.47	122.27	2.70	126.26	64.48	62.37	50.40	18.39	50.60	19.73	35.17	34.39	9.65	30.45	20.05
P2P-Bridge (Ours)	XYZ, RGB, DINO	**2.42**	**122.23**	**2.65**	**126.22**	**64.44**	**62.33**	**49.64**	18.57	**49.84**	19.92	**34.88**	**34.11**	9.57	**30.27**	**19.92**

4.6 Ablation Studies

We perform ablation studies to understand better the influence of network and diffusion parameter design choices on performance.

Diffusion Model Backbones. We evaluate different diffusion model backbones on a subset of ScanNet++. Specifically, we consider an architecture based on the Point-Voxel-Convolution neural network PVCNN [32] (*c.f.* Fig. 3), a transformer-based architecture from GECCO [55] and a sparse-convolution-based architecture using Minkowski Engine [7]. Since Minkowski Engine does not provide skeletons for diffusion model architectures, we recreate the DDPM [20] backbone architecture using only building blocks from Minkowski Engine. Table 4 shows the resulting performance, favoring the PVCNN architecture.

Bridge Settings. We evaluate the effect of nearest-neighbor interpolation and stochasticity on the PU-Net dataset. Table 3 shows that training without previous alignment of the unordered point cloud data drastically decreases the performance of our method. In fact, without proper data alignment, the method is unable to converge. Adding stochasticity to the interpolation path during training, which amounts to not training an OT-ODE but an SDE, also decreases performance. We speculate this is due to the strong prior information within noisy scans. However, for other tasks, such as point cloud completion, additional stochasticity could become necessary [30, 37].

Table 3: Bridge settings comparison on the PU-Net dataset. CD and P2M are multiplied by 10^4.

OT-ODE	Alignment	CD	P2M
✓	✗	49.33	44.22
✗	✓	2.45	0.73
✓	✓	**2.11**	**0.65**

Table 4: Diffusion model backbones comparison on ScanNet++. CD and P2M are both multiplied by 10^4.

Backbone	CD	P2M
Minkowksi [7]	12.79	32.39
SetTransformer [55]	10.50	15.46
PVCNN [32]	**9.76**	**14.33**

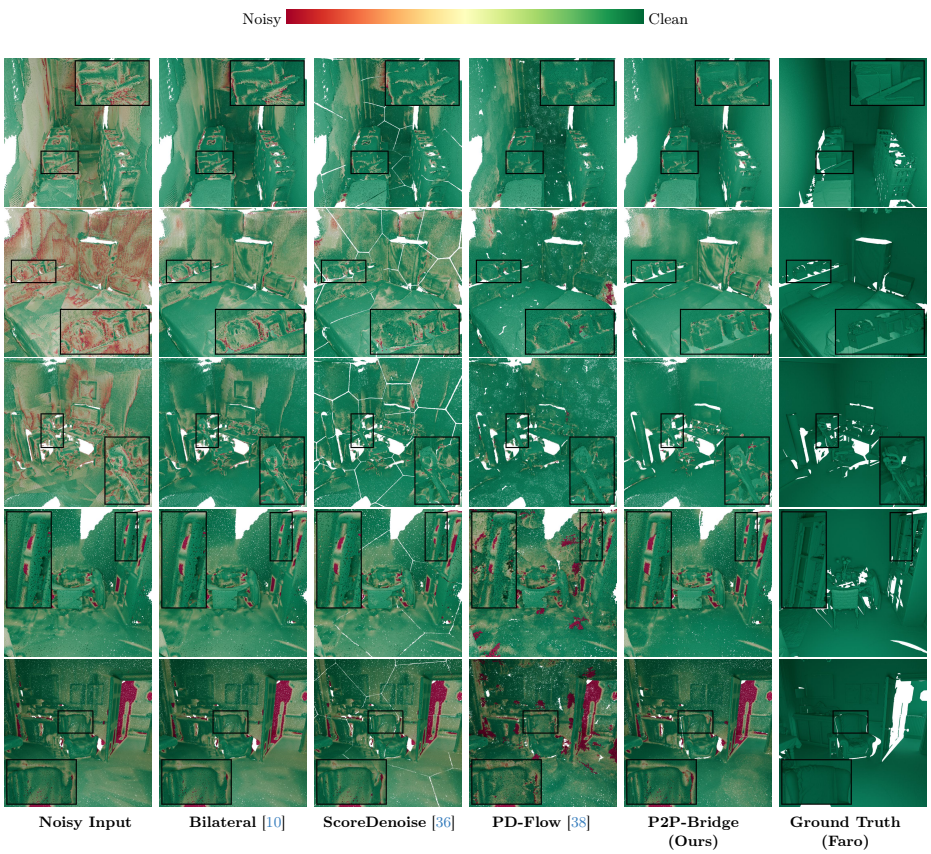

Fig. 5: Qualitative comparison on ScanNet++ [58] (top 3 rows) and ARKitScenes [1] (2 bottom rows) using noisy iPhone scans as input.

Model Architecture. We investigate the importance of individual building blocks and their attributes in Tab. 5. The study shows that increasing the number of blocks generally improves results, where the difference is larger for shallower blocks. Since the input is down-sampled after each SA-Block, shallower blocks can extract more fine-grained features, possibly explaining the larger impact, as the voxel convolutions and the global feature network already extract coarse features. Amongst additional feature layers, we see the biggest impact from the feature embedding. However, doubling the channels in each layer has the biggest impact on evaluation metrics. Figure 6 shows the relative change in metrics with increasing inference steps. Good results can be achieved with as little as five to ten inference steps, after which the metrics seem to plateau.

Table 5: Network configuration study on ScanNet++. SE describes squeeze-and-excitation blocks after convolutional layers, introduced in [22]. CD and P2M are multiplied by 10^4.

Base Channels	PVC Blocks	Global Feature	SE	Feature Embedding	CD (Δ)	P2M (Δ)
32	1222	✓	✓	✓	9.41 (+1.14)	13.78 (+4.08)
32	2122	✓	✓	✓	9.45 (+1.18)	13.82 (+4.12)
32	2212	✓	✓	✓	9.33 (+1.06)	13.70 (+4.00)
32	2221	✓	✓	✓	9.32 (+1.05)	13.70 (+4.00)
32	2222	✓	✓	✓	9.26 (+0.99)	13.67 (+3.97
32	2222	✗	✓	✓	9.36 (+1.09)	13.75 (+4.05)
32	2222	✓	✗	✓	9.33 (+1.06)	13.71 (+4.01)
32	2222	✓	✓	✗	9.76 (+1.49)	14.33 (+4.61)
64	2222	✓	✓	✓	8.31 (+0.04)	9.72 (+0.02)
64	2322	✓	✓	✓	**8.27** (−−)	**9.70** (−−)

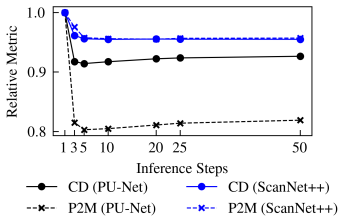

Fig. 6: Relative improvement over CD and P2M with increasing sampling steps. Good metrics can be achieved in as little as 5 to 10 steps, resulting in fast inference.

5 Conclusion and Discussion

In this paper, we presented P2P-Bridge, a point cloud denoising framework based on diffusion Schrödinger bridges. It approaches the denoising task as a data-to-data diffusion problem by learning an optimal transport path between point sets. We motivated the need for data alignment when applying diffusion bridges to point cloud data by drawing similarities between optimal transport plans and shortest-path point cloud interpolation and empirically show the efficiency of this approach. We applied our method on single object datasets as well as large-scale indoor point clouds and showed through extensive experiments that our method outperforms prior works on single objects as well as large point cloud data. Finally, we showed that additional image-based features, such as RGB information, as well as point-wise high-level features, such as DINOV2 features, further improve results.

Acknowledgements Francis Engelmann is partially supported by an ETH AI Center postdoctoral research fellowship and an ETH Zurich Career Seed Award.

References

1. Baruch, G., Chen, Z., Dehghan, A., Dimry, T., Feigin, Y., Fu, P., Gebauer, T., Joffe, B., Kurz, D., Schwartz, A., et al.: ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In: International Conference on Neural Information Processing Systems (NeurIPS) (2021) 8, 10, 12, 13
2. Chen, H., Shen, J.: Denoising of point cloud data for computer-aided design, engineering, and manufacturing. In: Engineering with Computers (2018) 2, 3, 10
3. Chen, H., Wei, M., Sun, Y., Xie, X., Wang, J.: Multi-patch collaborative point cloud denoising via low-rank recovery with graph constraint. In: IEEE transactions on visualization and computer graphics (2019) 3
4. Chen, T., Liu, G.H., Theodorou, E.A.: Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory (2022) 5, 6
5. Chen, Y., Hu, V.T., Gavves, E., Mensink, T., Mettes, P., Yang, P., Snoek, C.G.: PointMixup: Augmentation for Point Clouds. In: European Conference on Computer Vision (ECCV) (2020) 6
6. Chen, Z., He, G., Zheng, K., Tan, X., Zhu, J.: Schrodinger Bridges Beat Diffusion Models on Text-to-Speech Synthesis (2023) 5, 6, 7
7. Choy, C., Gwak, J., Savarese, S.: 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019) 12
8. De Bortoli, V., Thornton, J., Heng, J., Doucet, A.: Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In: International Conference on Neural Information Processing Systems (NeurIPS) (2021) 3, 5
9. Delitzas, A., Takmaz, A., Tombari, F., Sumner, R., Pollefeys, M., Engelmann, F.: SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2024) 1
10. Digne, J., de Franchis, C.: The Bilateral Filter for Point Clouds. In: Image Processing On Line (2017) 2, 9, 10, 12, 13
11. Digne, J., Valette, S., Chaine, R.: Sparse geometric representation through local shape probing. In: IEEE transactions on visualization and computer graphics (2017) 3
12. Duan, C., Chen, S., Kovacevic, J.: Weighted multi-projection: 3d point cloud denoising with tangent planes. In: 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (2018) 3
13. Duan, C., Chen, S., Kovacevic, J.: 3D point cloud denoising via deep neural network based local surface estimation (2019) 2
14. Engelmann, F., Manhardt, F., Niemeyer, M., Tateno, K., Tombari, F.: OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. In: International Conference on Learning Representations (ICLR) (2024) 1
15. Fei, B., Yang, W., Chen, W.M., Li, Z., Li, Y., Ma, T., Hu, X., Ma, L.: Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. In: IEEE Transactions on Intelligent Transportation Systems (2022) 11
16. Golub, G.H., Van Loan, C.F.: An analysis of the total least squares problem. In: SIAM journal on numerical analysis (1980) 3
17. Han, X.F., Jin, J.S., Wang, M.J., Jiang, W.: Guided 3d point cloud filtering. In: Multimedia Tools and Applications (2018) 2, 3

18. Han, X.F., Jin, J.S., Wang, M.J., Jiang, W., Gao, L., Xiao, L.: A review of algorithms for filtering the 3D point cloud. In: Signal Processing: Image Communication (2017) 2, 3
19. Hermosilla, P., Ritschel, T., Ropinski, T.: Total denoising: Unsupervised learning of 3D point cloud cleaning. In: Proceedings of the IEEE/CVF international conference on computer vision (2019) 2, 3
20. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: International Conference on Neural Information Processing Systems (NeurIPS) (2020) 3, 4, 5, 6, 12
21. Hoogeboom, E., Heek, J., Salimans, T.: simple diffusion: End-to-end diffusion for high resolution images. In: International Conference on Machine Learning (2023) 7
22. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2018) 7, 14
23. Hu, W., Gao, X., Cheung, G., Guo, Z.: Feature graph learning for 3d point cloud denoising. In: IEEE Transactions on Signal Processing (2020) 3
24. Huang, R., Peng, S., Takmaz, A., Tombari, F., Pollefeys, M., Song, S., Huang, G., Engelmann, F.: Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels. European Conference on Computer Vision (ECCV) (2024) 1
25. Kasten, Y., Rahamim, O., Chechik, G.: Point Cloud Completion with Pretrained Text-to-Image Diffusion Models. In: International Conference on Neural Information Processing Systems (NeurIPS) (2024) 4
26. Kreuzberg, L., Zulfikar, I.E., Mahadevan, S., Engelmann, F., Leibe, B.: 4D-StOP: Panoptic Segmentation of 4D LiDAR using Spatio-temporal Object Proposal Generation and Aggregation. In: ECCVW (2022) 1
27. Leal, E., Sanchez-Torres, G., Branch, J.W.: Sparse regularization-based approach for point cloud denoising and sharp features enhancement. In: Sensors (2020) 3
28. Lemke, O., Bauer, Z., Zurbrügg, R., Pollefeys, M., Engelmann, F., Blum, H.: Spot-Compose: A Framework for Open-Vocabulary Object Retrieval and Drawer Manipulation in Point Clouds. In: 2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024 (2024) 1
29. Léonard, C.: A survey of the schr\" odinger problem and some of its connections with optimal transport. In: arXiv preprint arXiv:1308.0215 (2013) 5
30. Liu, G.H., Vahdat, A., Huang, D.A., Theodorou, E.A., Nie, W., Anandkumar, A.: I2SB: Image-to-Image Schrödinger Bridge. In: International Conference on Machine Learning (ICML) (2023) 4, 5, 6, 7, 12
31. Liu, Z., Xiao, X., Zhong, S., Wang, W., Li, Y., Zhang, L., Xie, Z.: A feature-preserving framework for point cloud denoising. In: Computer-Aided Design (2020) 2, 3
32. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-Voxel CNN for Efficient 3D Deep Learning (2019) 7, 12
33. Lu, X., Schaefer, S., Luo, J., Ma, L., He, Y.: Low rank matrix approximation for 3D geometry filtering. In: IEEE Transactions on Visualization and Computer Graphics (2020) 3
34. Luo, S., Hu, W.: Differentiable Manifold Reconstruction for Point Cloud Denoising (2020) 2, 3, 9, 10, 12
35. Luo, S., Hu, W.: Diffusion Probabilistic Models for 3D Point Cloud Generation. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4
36. Luo, S., Hu, W.: Score-Based Point Cloud Denoising. In: International Conference on Computer Vision (ICCV) (2021) 2, 3, 4, 9, 10, 11, 12, 13

37. Lyu, Z., Kong, Z., XU, X., Pan, L., Lin, D.: A Conditional Point Diffusion-Refinement Paradigm for 3D Point Cloud Completion. In: International Conference on Learning Representations (ICLR) (2022) 4, 7, 11, 12
38. Mao, A., Du, Z., Wen, Y.H., Xuan, J., Liu, Y.J.: PD-Flow: A point cloud denoising framework with normalizing flows. In: European Conference on Computer Vision (ECCV) (2022) 2, 3, 9, 10, 11, 12, 13
39. Mattei, E., Castrodad, A.: Point cloud denoising via moving RPCA. In: Computer Graphics Forum (2017) 3
40. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning (ICML) (2021) 4
41. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: DINOv2: Learning Robust Visual Features without Supervision (2023) 2, 3, 8
42. Peyré, G., Cuturi, M.: Computational optimal transport. Foundations and Trends in Machine Learning (2019) 7
43. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: International Conference on Neural Information Processing Systems (NeurIPS) (2017) 7
44. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies. In: International Conference on Neural Information Processing Systems (NeurIPS) (2022) 2
45. Rakotosaona, M.J., La Barbera, V., Guerrero, P., Mitra, N.J., Ovsjanikov, M.: PointCleanNet: Learning to Denoise and Remove Outliers from Dense Point Clouds (2020) 2, 3, 4, 8, 9, 10
46. Ravi, N., Reizenstein, J., Novotny, D., Gordon, T., Lo, W.Y., Johnson, J., Gkioxari, G.: Accelerating 3d deep learning with pytorch3d. In: arXiv preprint arXiv:2007.08501 (2020) 2
47. Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., Norouzi, M.: Palette: Image-to-image diffusion models. In: ACM SIGGRAPH (2022) 4
48. Schrödinger, E.: Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. Annales de l'institut Henri Poincaré (1932) 3, 5
49. de Silva Edirimuni, D., Lu, X., Shao, Z., Li, G., Robles-Kelly, A., He, Y.: Iterativepfn: True iterative point cloud filtering. In: International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13530–13539 (2023) 3, 10, 12
50. Somnath, V.R., Pariset, M., Hsieh, Y.P., Martinez, M.R., Krause, A., Bunne, C.: Aligned Diffusion Schr\" odinger Bridges. In: arXiv preprint arXiv:2302.11419 (2023) 5
51. Song, J., Meng, C., Ermon, S.: Denoising Diffusion Implicit Models. In: International Conference on Learning Representations (ICLR) (2021) 4, 6
52. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: International Conference on Neural Information Processing Systems (NeurIPS) (2019) 3, 11
53. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2023) 1
54. Takmaz, A., Schult, J., Kaftan, I., Akçay, M., Leibe, B., Sumner, R., Engelmann, F., Tang, S.: 3D Segmentation of Humans in Point Clouds with Synthetic Data. In: International Conference on Computer Vision (ICCV) (2023) 1
55. Tyszkiewicz, M.J., Fua, P., Trulls, E.: Gecco: Geometrically-conditioned point diffusion models. arXiv preprint arXiv:2303.05916 (2023) 4, 12

56. Weder, S., Schonberger, J., Pollefeys, M., Oswald, M.R.: RoutedFusion: Learning Real-time Depth Map Fusion. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
57. Yadav, S.K., Reitebuch, U., Skrodzki, M., Zimmermann, E., Polthier, K.: Constraint-based point set denoising using normal voting tensor and restricted quadratic error metrics. In: Computers & Graphics (2018) 2, 3
58. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In: International Conference on Computer Vision (ICCV) (2023) 8, 10, 12, 13
59. Yilmaz, K., Schult, J., Nekrasov, A., Leibe, B.: MASK4D: Mask Transformer for 4D Panoptic Segmentation (2024) 1
60. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: PU-Net: Point Cloud Upsampling Network. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 8, 10
61. Yue, Y., Das, A., Engelmann, F., Tang, S., Lenssen, J.: Improving 2D Feature Representations by 3D-Aware Fine-Tuning. European Conference on Computer Vision (ECCV) (2024) 2
62. Yue, Y., Kontogianni, T., Schindler, K., Engelmann, F.: Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 1
63. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. In: CVPR (2017) 4, 8, 10, 12
64. Zeng, J., Cheung, G., Ng, M., Pang, J., Yang, C.: 3D Point Cloud Denoising using Graph Laplacian Regularization of a Low Dimensional Manifold Model (2019) 3, 9, 10
65. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: LION: Latent Point Diffusion Models for 3D Shape Generation. In: International Conference on Neural Information Processing Systems (NeurIPS) (2022) 4, 7
66. Zhang, D., Lu, X., Qin, H., He, Y.: Pointfilter: Point cloud filtering via encoder-decoder modeling. In: IEEE Transactions on Visualization and Computer Graphics (2020) 2
67. Zhang, F., Zhang, C., Yang, H., Zhao, L.: Point Cloud Denoising With Principal Component Analysis and a Novel Bilateral Filter. In: Traitement du signal (2019) 3
68. Zhao, Y., Zheng, H., Wang, Z., Luo, J., Lam, E.Y.: Point Cloud Denoising via Momentum Ascent in Gradient Fields (2023) 2, 3, 4, 9, 10, 11, 12
69. Zheng, J., Barath, D., Pollefeys, M., Armeni, I.: Map-adapt: Real-time quality-adaptive semantic 3d maps. arXiv preprint arXiv:2406.05849 (2024) 4
70. Zheng, Y., Li, G., Wu, S., Liu, Y., Gao, Y.: Guided point cloud denoising via sharp feature skeletons. In: The Visual Computer (2017) 2, 3
71. Zheng, Y., Li, G., Xu, X., Wu, S., Nie, Y.: Rolling normal filtering for point clouds. In: Computer Aided Geometric Design (2018) 2, 3
72. Zhou, L., Du, Y., Wu, J.: 3D Shape Generation and Completion Through Point-Voxel Diffusion. In: International Conference on Computer Vision (ICCV) (2021) 4, 7
73. Zhu, Z., Peng, S., Larsson, V., Cui, Z., Oswald, M.R., Geiger, A., Pollefeys, M.: NICER-SLAM: Neural Implicit Scene Encoding for RGB SLAM. In: International Conference on 3D Vision (3DV) (2024) 4
74. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4

75. Zurbrügg, R., Liu, Y., Engelmann, F., Kumar, S., Hutter, M., Patil, V., Yu, F.: ICGNet: A Unified Approach for Instance-Centric Grasping. In: International Conference on Robotics and Automation (ICRA) (2024) 1