Diffusion models for denoising point clouds

Authors
Jonathan Cederlund, Hanxuan Lin

June 5, 2025

Summary

Motivation: Reconstruction problem

- Classical robotics problem.
- Accurate 3D reconstruction is essential for spatial understanding.
- Sensor noise and environmental interference degrade point clouds.

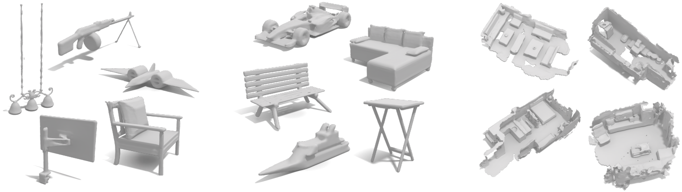


Figure 1: Examples of 3D reconstruction [Williams et al., 2021]

Motivation: Point Cloud Denoising

- Noisy reconstructions impair downstream tasks like recognition, and planning.
- Denoising is critical for robust robotic perception.
- We evaluate **P2P-Bridge**, a diffusion-based denoising model.

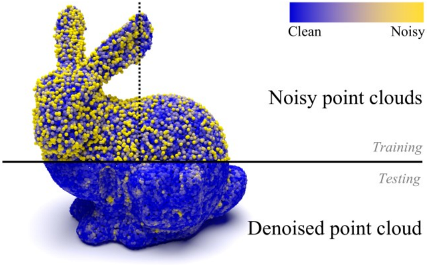


Figure 2: Noisy vs. denoised point clouds [Wang et al., 2024]

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.
- **Deep learning methods:** Learn point-wise displacements,

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.
- **Deep learning methods:** Learn point-wise displacements,
  - Iterative-PFN

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.
- **Deep learning methods:** Learn point-wise displacements,
  - Iterative-PFN
  - PD-Flow

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.
- **Deep learning methods:** Learn point-wise displacements,
  - Iterative-PFN
  - PD-Flow
  - **Diffusion models:** Gradually denoise in steps.

Previous Work

- **Signal-based methods:** Outliers removal using low-pass filters.
- **Optimization-based methods:** Local neighborhood optimization.
- **Deep learning methods:** Learn point-wise displacements,
  - Iterative-PFN
  - PD-Flow
  - **Diffusion models:** Gradually denoise in steps.
    - P2P-Bridge

Problem formulation

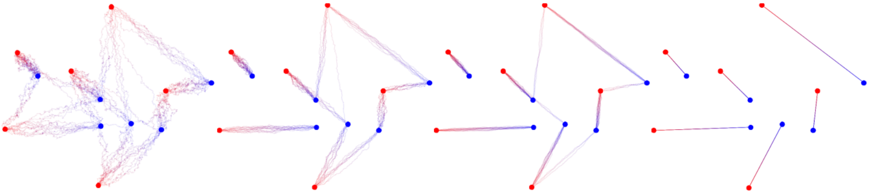


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Problem formulation

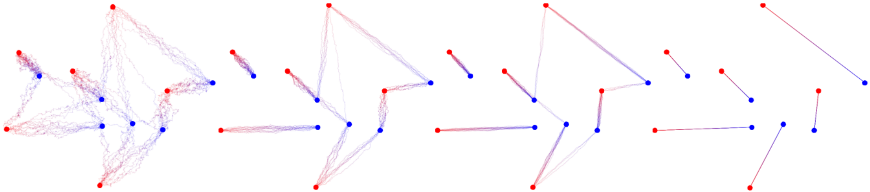


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Definitions

- Clean point cloud $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M \times 3}$ (Blue points)

Problem formulation

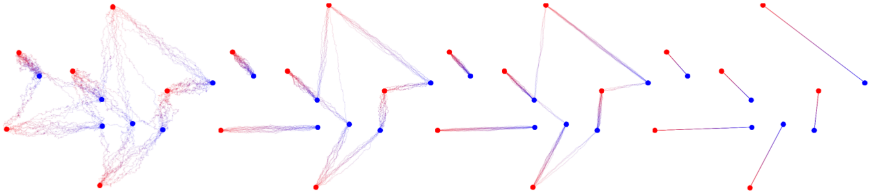


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Definitions

- Clean point cloud $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M\times 3}$ (Blue points)
- Noisy point cloud $\tilde{\mathcal{P}} = \{\tilde{x}_i\} \in \mathbb{R}^{N\times 3}$ (Red points)

Problem formulation

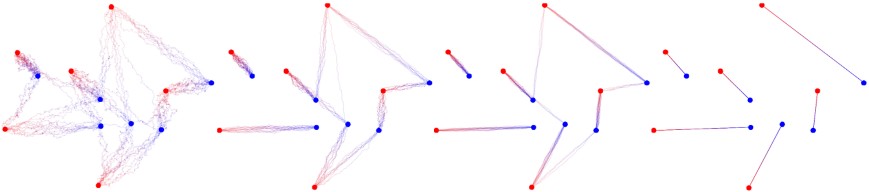


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Definitions

- Clean point cloud $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M \times 3}$ (Blue points)
- Noisy point cloud $\tilde{\mathcal{P}} = \{\tilde{x}_i\} \in \mathbb{R}^{N \times 3}$ (Red points)
- Model $f_\theta(x)$

Problem formulation

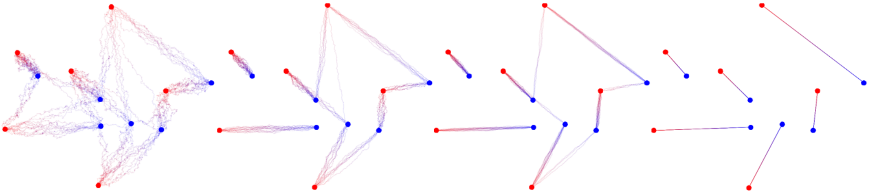


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Definitions

- Clean point cloud $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M\times 3}$ (Blue points)
- Noisy point cloud $\tilde{\mathcal{P}} = \{\tilde{x}_i\} \in \mathbb{R}^{N\times 3}$ (Red points)
- Model $f_\theta(x)$
- Diffusion step $x_{t+1} = x_t + f_\theta(x_t)$

Problem formulation

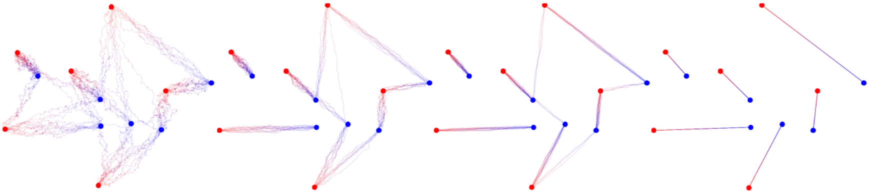


Figure 3: Visualization of optimal transport[Peyré and Cuturi, 2020]

Definitions

- Clean point cloud $\mathcal{P} = \{x_i\} \in \mathbb{R}^{M \times 3}$ (Blue points)
- Noisy point cloud $\tilde{\mathcal{P}} = \{\tilde{x}_i\} \in \mathbb{R}^{N \times 3}$ (Red points)
- Model $f_\theta(x)$
- Diffusion step $x_{t+1} = x_t + f_\theta(x_t)$
- Kullback Leibler divergence (KL)

Schrödingers bridge

SDE

$$\mathrm{d}\mathbf{x}_t^1 = \left[\mathbf{f}(\mathbf{x}_t^1, t)\,\mathrm{d}t + g^2(t)\nabla \log \Psi_t(\mathbf{x}_t^1)\right]\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \mathbf{x}_0 \sim p_{\text{data}} \tag{1}$$

$$\mathrm{d}\mathbf{x}_t^2 = \left[\mathbf{f}(\mathbf{x}_t^2, t)\,\mathrm{d}t - g^2(t)\nabla \log \hat{\Psi}_t(\mathbf{x}_t^2)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t \quad \mathbf{x}_t \sim p_{\text{prior}} \tag{2}$$

[Vogel et al., 2024]

Definitions

- w_t is a Wiener process. (White noise)

Schrödingers bridge

SDE

$$\mathrm{d}\mathbf{x}_t^1 = \left[\mathbf{f}(\mathbf{x}_t^1, t)\,\mathrm{d}t + g^2(t)\nabla \log \Psi_t(\mathbf{x}_t^1)\right]\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \mathbf{x}_0 \sim p_{\text{data}} \quad (1)$$

$$\mathrm{d}\mathbf{x}_t^2 = \left[\mathbf{f}(\mathbf{x}_t^2, t)\,\mathrm{d}t - g^2(t)\nabla \log \hat{\Psi}_t(\mathbf{x}_t^2)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t \quad \mathbf{x}_t \sim p_{\text{prior}} \quad (2)$$

[Vogel et al., 2024]

Definitions

- w_t is a Wiener process. (White noise)
- $f(x,t)$ is a vector-valued function known as the drift.

Schrödingers bridge

SDE

$$\mathrm{d}\mathbf{x}_t^1 = \left[\mathbf{f}(\mathbf{x}_t^1, t)\,\mathrm{d}t + g^2(t)\nabla \log \Psi_t(\mathbf{x}_t^1)\right]\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \mathbf{x}_0 \sim p_{\mathrm{data}} \tag{1}$$

$$\mathrm{d}\mathbf{x}_t^2 = \left[\mathbf{f}(\mathbf{x}_t^2, t)\,\mathrm{d}t - g^2(t)\nabla \log \hat{\Psi}_t(\mathbf{x}_t^2)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t \quad \mathbf{x}_t \sim p_{\mathrm{prior}} \tag{2}$$

[Vogel et al., 2024]

Definitions

- w_t is a Wiener process. (White noise)
- $f(x,t)$ is a vector-valued function known as the drift.
- $g(t)$ is a scalar-valued term referred to as the diffusion coefficient.

Schrödingers bridge

SDE

$$\mathrm{d}\mathbf{x}_t^1 = \left[\mathbf{f}(\mathbf{x}_t^1, t)\,\mathrm{d}t + g^2(t)\nabla \log \Psi_t(\mathbf{x}_t^1)\right]\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t \quad \mathbf{x}_0 \sim p_{\text{data}} \tag{1}$$

$$\mathrm{d}\mathbf{x}_t^2 = \left[\mathbf{f}(\mathbf{x}_t^2, t)\,\mathrm{d}t - g^2(t)\nabla \log \hat{\Psi}_t(\mathbf{x}_t^2)\right]\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t \quad \mathbf{x}_t \sim p_{\text{prior}} \tag{2}$$

[Vogel et al., 2024]

Definitions

- w_t is a Wiener process. (White noise)
- $f(x, t)$ is a vector-valued function known as the drift.
- $g(t)$ is a scalar-valued term referred to as the diffusion coefficient.
- $\nabla \log \Psi_t(\mathbf{x}_t)$ and $\nabla \log \hat{\Psi}_t(\mathbf{x}_t)$ are additional nonlinear drift terms

Examples of Schödingers Bridge

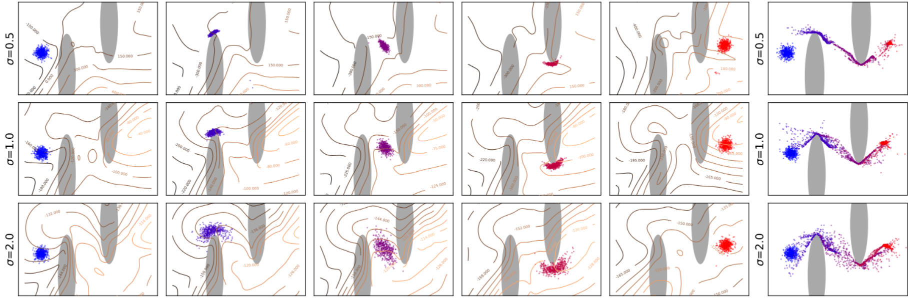


Figure 4: Step-wise example of SB minimizing the transport cost.

[Liu et al., 2022]

Evaluation Metric: Chamfer Distance

Defenition

$$\mathrm{CD}\big(\hat{\mathcal{P}}, \mathcal{P}\big) \;=\; \frac{1}{2n}\sum_{i=1}^{n}\|\hat{\mathbf{x}}_i - \mathrm{NN}(\hat{\mathbf{x}}_i, \mathcal{P})\|_2^2 + \frac{1}{2m}\sum_{j=1}^{m}\|\mathbf{x}_j - \mathrm{NN}(\mathbf{x}_j, \hat{\mathcal{P}})\|_2^2 \quad (3)$$

Key Features

- Measures proximity between predicted and ground-truth point sets in both directions.
- Penalizes both noise (outliers) and missing regions.

Evaluation Metric: Point-to-Mesh Distance (P2M)

Definition

$$\mathrm{P2M}\big(\widehat{\mathcal{P}}, \mathcal{M}\big) = \underbrace{\frac{1}{2n} \sum_{i=1}^{n} \min_{f \in \mathcal{F}} d(\widehat{\mathbf{x}}_i, f)}_{\text{Point}\to\text{Face (P2F)}} + \underbrace{\frac{1}{2|\mathcal{F}|} \sum_{f \in \mathcal{F}} \min_{\widehat{\mathbf{x}}_i \in \widehat{\mathcal{P}}} d(\widehat{\mathbf{x}}_i, f)}_{\text{Face}\to\text{Point (F2P)}} \tag{4}$$

Key Features

- **P2F:** Measures point accuracy on surface.
- **F2P:** Checks for complete surface coverage.
- Less sensitive to sampling density, geometry-aware.

Experiments

Datasets

- PU-Net
- Our dataset

Experimental details

- CD and P2M on gaussian noise
- Quantitative and qualitative analysis of point cloud collapse
- Extending the diffusion process past what the model is trained for.

Qualitative comparison

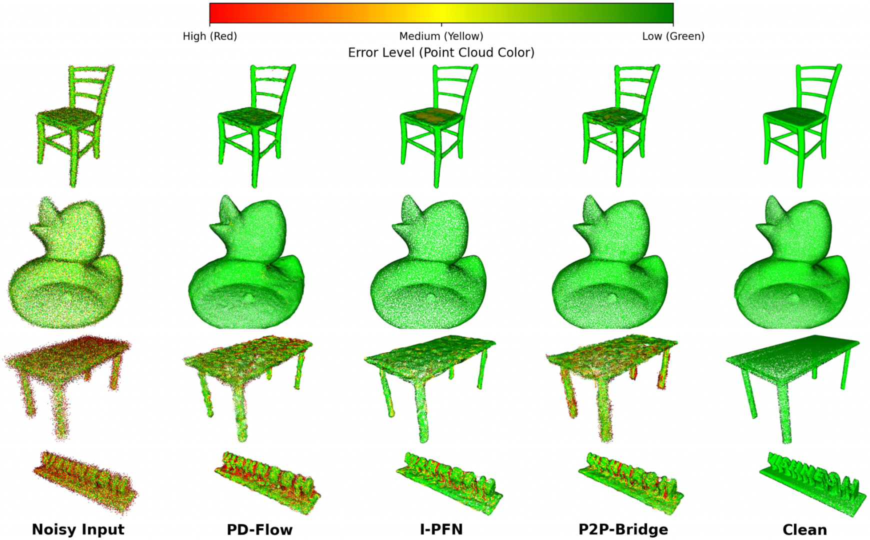


Figure 5: Qualitative comparison of various point cloud denoising methods against P2P-Bridge on PU-Net and our own generated dataset under isotopic

Stepwise Evaluation of Denoising Performance

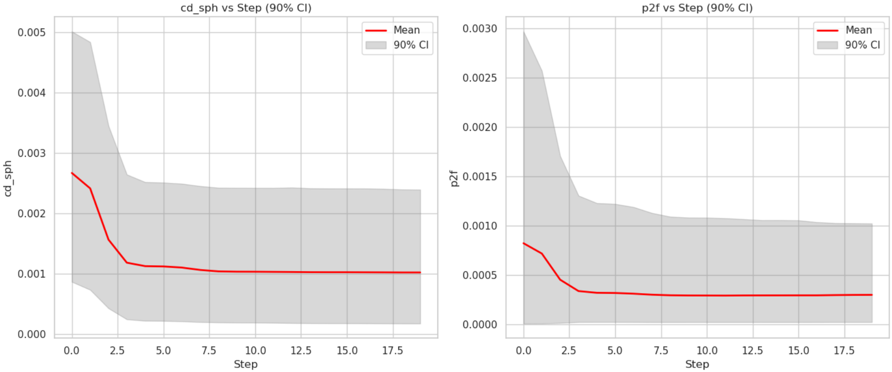


Figure 6: Chamfer Distance and Point-to-Mesh (P2M) values over optimization steps with 90% confidence intervals on our own dataset with 3% isotopic Gaussian noise.

Stepwise Evaluation of Denoising Performance

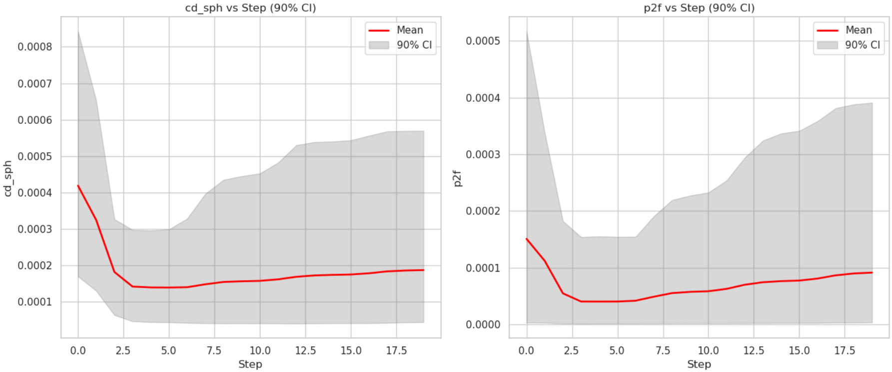
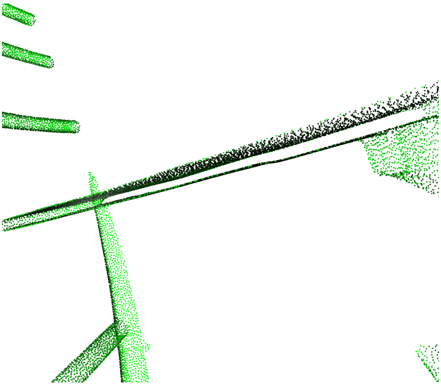
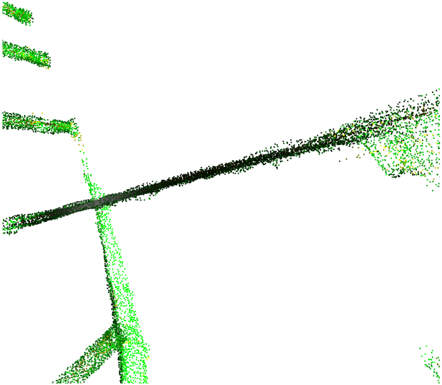


Figure 7: Chamfer Distance and Point-to-Mesh (P2M) values over optimization steps with 90% confidence intervals on our own dataset with 1% isotopic Gaussian noise.

Point cloud collapse





Object-level Denoising Results

	Method	$10 \cdot 10^3$ (sparse)									$50 \cdot 10^3$ (dense)								
		CD	P2M	In %	CD	P2M	In %	CD	P2M	In %	CD	P2M	In %	CD	P2M	In %	CD	P2M	In %
Our Dataset		3%			6%			9%			3%			6%			9%		
	PD-Flow [16]	4.32	1.15	54.9	11.21	3.32	43.2	33.04	9.51	34.5	1.99	0.59	51.1	10.90	3.31	36.9	25.07	6.96	30.5
	I-PFN [5]	3.68	1.08	55.2	9.66	2.99	40.2	25.29	7.80	33.4	1.30	0.30	49.0	7.65	2.36	35.8	16.51	4.95	34.3
	P2P-B [23]	4.10	1.21	55.2	6.29	1.99	48.9	12.21	3.90	43.0	1.39	0.40	53.1	4.82	1.24	41.4	11.22	3.18	35.8
PUNet [15]		1%			2%			3%			1%			2%			3%		
	PD-Flow [16]	2.13	0.38	55.5	3.25	1.01	54.9	5.19	2.52	50.6	0.65	0.16	56.6	1.42	0.78	54.2	3.90	2.86	48.7
	I-PFN [5]	2.31	0.37	51.8	3.43	0.90	51.4	5.49	2.50	43.9	0.66	0.12	50.3	1.05	0.43	50.8	2.54	1.65	41.1
	P2P-B [23]	2.28	0.39	56.4	3.20	0.81	55.0	3.99	1.42	53.3	0.59	0.09	54.2	0.90	0.32	52.9	1.56	0.84	50.0

Figure 8: Quantitative comparison of Chamfer Distance (CD) and Point-to-Mesh (P2M) distance metrics, evaluated on PU-Net and our own generated dataset under varying levels of isotropic Gaussian noise.

Discussion

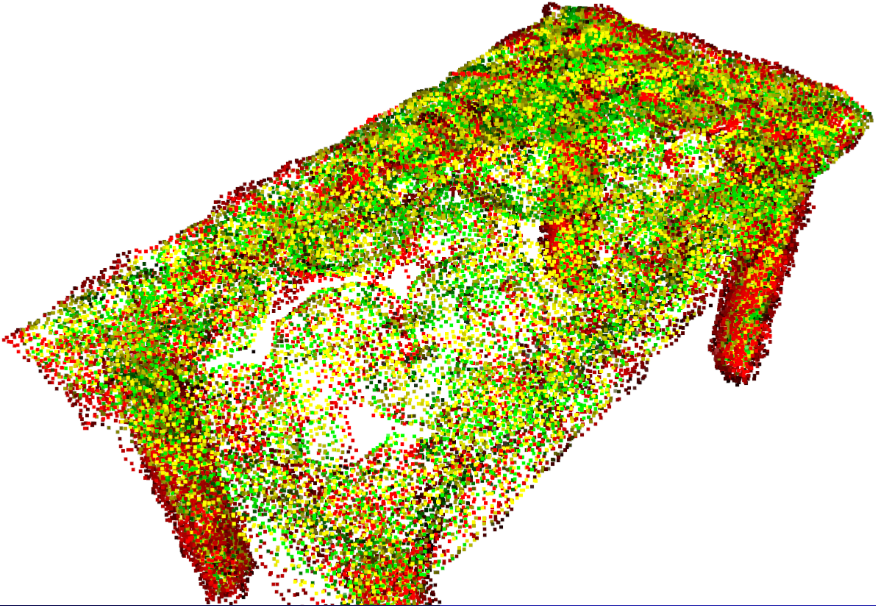
Future directions

Real-time inference

Limitations

- Hardware limitation
  - Computation complexity
  - Ideally run on device or local setting
- Software limitation
  - Point cloud reconstruction using RealSense camera

Limitations



Citations

- Liu, G.-H., Chen, T., So, O., and Theodorou, E. A. (2022).
  Schrödinger bridge.
- Peyré, G. and Cuturi, M. (2020).
  Computational optimal transport.
- Vogel, M., Tateno, K., Pollefeys, M., Tombari, F., Rakotosaona, M.-J., and Engelmann, F. (2024).
  P2p-bridge: Diffusion bridges for 3d point cloud denoising.
- Wang, W., Liu, X., Zhou, H., Wei, L., Deng, Z., Murshed, M., and Lu, X. (2024).
  Noise4denoise: Leveraging noise for unsupervised point cloud denoising.
  Computational Visual Media, 10:659–669.
- Williams, F., Gojcic, Z., Khamis, S., Zorin, D., Bruna, J., Fidler, S., and Litany, O. (2021).
  Neural fields as learnable kernels for 3d reconstruction.