PUFM: Efficient Point Cloud Upsampling via Flow Matching

Zhi-Song Liu[1], **Chenhang He**[2*], **Yakun Ju**[3], **Lei Li**[4, 5†]
[1]Department of Computational Engineering, Lappeenranta-Lahti University of Technology
[2]The Hong Kong Polytechnic University
[3]School of Computing and Mathematical Sciences, University of Leicester
[4]Technical University of Munich
[5]University of Virginia
zhisong.liu@lut.fi, chenhang.he@polyu.edu.hk, yj174@leicester.ac.uk, leili@virginia.edu

Abstract

Diffusion models have recently been adopted for point cloud upsampling due to their effectiveness in solving ill-posed problems. However, existing upsampling methods often struggle with inefficiencies, as they generate dense point clouds by mapping Gaussian noise to data, overlooking the geometric information already present in sparse inputs. To address this, we propose PUFM, a novel **P**oint **C**loud **U**psampling via **F**low **M**atching, which learns to directly transform sparse point clouds into their high-fidelity dense counterparts. Our approach first applies midpoint interpolation to densify the sparse input. Then, we construct a continuous interpolant between sparse and dense point clouds and train a neural network to estimate the velocity field for flow matching. Given the unordered nature of point clouds, we introduce a pre-alignment step based on Earth Mover's Distance (EMD) optimization to ensure coherent and meaningful interpolation between sparse and dense representation. This results in a more stable and efficient learning trajectory during flow matching. Experiments on synthetic benchmarks demonstrate that our method delivers superior upsampling quality but with fewer sampling steps. Further experiments on ScanNet and KITTI also show that our approach generalizes well to real-world RGB-D and LiDAR point clouds, making it more practical for real-world applications.

Code — https://github.com/Holmes-Alan/PUFM

Introduction

Point cloud upsampling (Yu and et al. 2018) has gained significant attention recently, aiming to increase point cloud resolution and enrich geometric details. Higher-resolution point clouds greatly benefit downstream tasks such as mesh reconstruction (Hanocka and et al. 2020), 3D scene perception (Yang and et al. 2024; Lemke and et al. 2024), and rendering (Kerbl and et al. 2023).

Point cloud upsampling is a fundamentally ill-posed problem. Conventional methods (Singh and et al. 2007; L. and et al. 2007) often rely on computationally intensive optimization procedures and handcrafted priors. However, these approaches typically struggle in challenging scenarios, particularly when only sparse and irregular inputs are available, making it difficult to recover the underlying surface topology. Recent deep learning-based methods (Yifan and et al. 2019; He and et al. 2023; Qu and et al. 2024; Li et al. 2019; Rong and et al. 2024) have shown superior performance by leveraging data-driven priors. These methods train deep networks to learn rich geometric representations from sparse point sets, enabling accurate and topology-aware upsampling to high-resolution point clouds. Among these, PUNet (Yu and et al. 2018) is a pioneering work that learns multi-level features per point and reconstructs the underlying surface based on the contextual information.

Recent advances in diffusion models (Ho and et al. 2020) for solving ill-posed problems has led to their applications to point cloud upsampling. The PUDM method (Qu and et al. 2024) proposed a conditional probability diffusion model that treats point cloud upsampling as a noise-to-data generative process, conditioned on sparse inputs. This approach has demonstrated high-quality dense prediction for both uni-

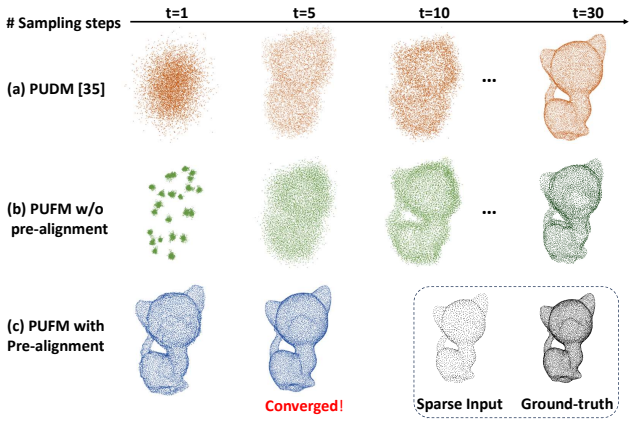

Figure 1: Convergence comparison between different distribution mapping paths. The diffusion model PUDM (Qu and et al. 2024) converges slowly since it starts from a noise distribution. In contrast, our proposed PUFM learns flow matching directly from sparse to dense point clouds, resulting in a more efficient upsampling process.

*Corresponding author.
†L. Li has been with UVA since August 2025.

form and nonuniform point clouds. However, we argue that this noise-to-data process is inefficient for point cloud upsampling, as sparse point cloud inputs already encode valuable prior information about the target structure. As illustrated in Figure 1a, PUDM converges slowly in predicting dense point clouds, indicating that this noise-to-data process may introduce unnecessary complexity for this task.

In this work, we propose **Point cloud Upsampling via Flow Matching (PUFM)**, an efficient method that directly learns the optimal transport between the underlying distributions of sparse and dense point clouds, significantly reducing both the learning complexity and sampling cost compared to diffusion models. To address the cardinality difference between the two distributions, we first apply midpoint interpolation (He and et al. 2023) to densify the sparse point clouds. In addition, we observe that the flow matching model encounters ambiguous learning during the early stages, as indicated by the collapsed ball-shaped clusters in Figure 1b. This issue arises from the inherent unordered nature of point clouds, which leads to multiple bijective matching paths between the sparse and dense point sets. Due to this matching ambiguity, the flow matching model tends to learn a suboptimal "average path" during the early steps of training, which hinders its convergence. To mitigate this, we propose to pre-align the sparse point clouds to their corresponding ground truth by optimizing their Earth Mover's Distance (EMD). This pre-alignment strategy mitigates the ambiguity in the flow matching process by enforcing clearer correspondences between sparse and dense point distributions. As a result, the learning trajectory becomes significantly smoother, leading to faster convergence during training. Notably, this pre-alignment is only applied during training and is not required at inference time. As shown in Figure 1c, our flow matching model predicts high-fidelity dense point clouds with a few sampling steps, achieving high accuracy and efficiency.

We evaluate the effectiveness of PUFM on datasets including PUGAN (Li et al. 2019) and PU1K (Qian and et al. 2021), where PUFM achieves state-of-the-art performance. Furthermore, we extend our method to real-world datasets, including Scannet (Dai et al. 2017) and KITTI (Geiger and et al. 2013). The qualitative results highlight our method's ability to upsample extremely sparse instances, showing its superior generalization and robustness in diverse scenarios.

Related Work

Point Cloud Analysis

Different from image data, point clouds contain irregular structures and are invariant to permutation. PointNet (Charles et al. 2017) is the pioneering work that applies shared MLPs to extract point-wise features, and then uses maxpooling for the extraction of permutation-invariant features. PointNet++ (Qi and et al. 2017) further improves it by proposing set abstraction to extract multi-level features. Similar works can be also found in (Ma and et al. 2022; Qian and et al. 2024; Wang and et al. 2019; Zhang and et al. 2019; Thomas and et al. 2019). For instance, PVCNN (Lu and et al. 2022) modifies the PointNet++ to handle vocalized point cloud representation, and a low rank matrix approximation algorithm is proposed to estimate normals for underlying 3D surface reconstruction. Recently, attention mechanisms have been widely explored in point cloud processing. Models such as Point Cloud Transformer (PCT) (Guo 2021), its variants PCT2 (Wu and et al. 2022) and FPCT (Park and et al. 2022), as well as Point Transformer (PT) (Zhao et al. 2021), Point Transformer V2 (PTV2) (Wu et al. 2022), and PTV3 (Wu and et al. 2024), have demonstrated impressive performance in tasks like point cloud recognition, denoising, and enhancement. These models leverage attention to capture complex relationships within point clouds, leading to more expressive and robust feature representations.

In this work, we develop a flow matching model for point cloud upsampling, leveraging PointNet++'s capability to encode multi-level contextual information, which makes it well-suited for learning accurate flow fields.

Point Cloud Upsampling

Point cloud upsampling has been extensively studied in recent years, with several approaches proposed to enhance point cloud resolution and improve geometric details. Early learning-based methods, such as PUNet (Yu and et al. 2018), paved the way for using deep learning in point cloud upsampling. MPU (Yifan and et al. 2019) introduced a progressive upsampling technique using patch-based processing, while PUGCN (Qian and et al. 2021) employed graph convolution networks for learning local point information via NodeShuffle. PUBP (Liu and et al. 2023) introduced a back-projection network for iterative down- and up-sampling refinement, and PUGAN (Li et al. 2019) leveraged adversarial networks to supervise the generation distribution, achieving more evenly distributed dense points. Further developments include self-supervised learning methods (Wenbo and et al. 2022; Metzer and et al. 2021) for exploiting local-scale geometric recurrence, as well as methods like PUGeo (Qian and et al. 2020) and NePs (Feng and et al. 2022) that project point clouds to 2D for continuous upsampling via convolution. Other works (Li, Pang, and Wang 2024; Chen and et al. 2023) focus on combining upsampling and denoising with joint supervision to better integrate point features. Despite these advances, most methods still rely on Chamfer distance, which fails to capture detailed underlying 3D structures.

In this paper, we propose a flow matching framework for point cloud upsampling that directly models the transport between sparse and dense point cloud distributions. Unlike prior methods that rely on Chamfer distance or displacement-based objectives (Yifan and et al. 2019; Li and et al. 2021; Rong and et al. 2024; Du et al. 2025), our approach efficiently learns the mapping by optimizing the l_2 loss along an iteratively constructed inverse path. While related to recent advances in score-based denoising models (Qu and et al. 2024; Kumbar and et al. 2023) and normalizing flows (Mao et al. 2023), which generate dense point clouds from noise, our method differs fundamentally by directly conditioning on sparse input, resulting in significantly faster and more efficient inference.

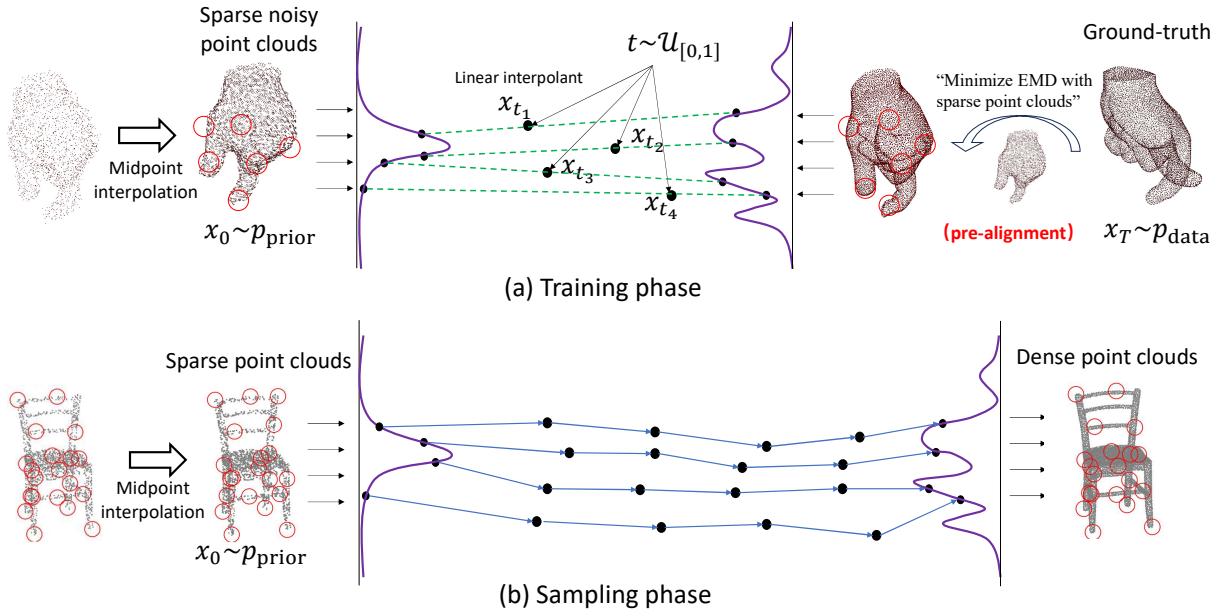

Figure 2: Illustration of the flow matching process for point cloud upsampling. (a) Training phase: Sparse noisy point clouds are processed with midpoint interpolation and linearly interpolated to match ground-truth dense point clouds with pre-alignment to minimize Earth Mover's Distance (EMD), where the red circles emphasize the patch as the unit for flow matching. (b) Sampling phase: Sparse point clouds with midpoint interpolation are transformed into dense point clouds using learned trajectories.

Approach

Preliminary

Given the target data $x \sim p(x)$, we aim to build a flow matching model (Lipman and et al. 2023) to learn a time-dependent process that transforms an initial distribution $p_0(x)$ at $t = 0$ into the target distribution $p(x)$ at $t = 1$:

$$x_t = \alpha_t x_0 + \sigma_t x_1, \text{where} \quad t \in [0,1], x_0 \sim p_0(x), x_1 \sim p(x), \tag{1}$$

where α_t is an increasing function of t satisfying $\alpha_0 = 1$, $\alpha_1 = 0$, and σ_t is a decreasing function of t satisfying $\sigma_0 = 0, \sigma_1 = 1$. The marginal probability distribution $p_t(x)$ evolves to satisfy $p_0(x)$ at $t = 0$ and $p_1(x) \approx p(x)$ at $t = 1$.

Flow matching formulates this as solving the following continuity equation:

$$\frac{dp_t(x)}{dt} + \nabla \cdot (p_t(x)\nu_\theta(x,t)) = 0, \tag{2}$$

where $\nu_\theta(x,t)$ is a parameterized velocity field that governs the evolution of $p_t(x)$.The velocity field $\nu_\theta(x,t)$ is learned by minimizing the following objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[\|(\sigma_t x_1 - \alpha_t x_0) - \nu_\theta(x_t,t)\|^2\right] \tag{3}$$

This objective ensures that the learned velocity aligns with the transport direction from x_0 to x_1. Once $\nu_\theta(x,t)$ is learned, samples from $p_1(x) \approx p(x)$ can be generated by integrating the ODE from $t = 0$ to $t = 1$, starting from a sample $x_0 \sim p_0(x)$. This can be approximated using Euler integration:

$$x_{t+\Delta} = x_t + (\delta_{t+\Delta} - \delta_t)\nu_\theta(x_t,t), \tag{4}$$

where δ represents the discrete timestep.

Point cloud Upsampling via Flow Matching

We propose to leverage flow matching for point cloud upsampling. One related work is PUDM (Qu and et al. 2024), which learns a straightforward noise-to-data model, and then directly optimizes the l_2 distance between the true and estimated point clouds. This approach has two limitations: 1) *sparse point clouds already contain partial views of the dense points, which encodes the underlying structure of the dense point clouds. Learning from noise is computationally costly and inefficient*; and 2) *Point clouds are unordered and have an irregular format, making it challenging for the model to pair points for learning tractable interpolation.* To resolve these issues, we propose a flow matching model PUFM for point cloud upsampling, as shown in Figure 2, which directly learns the optimal transport from sparse to dense distribution. To learn the flow matching model, we first densify the sparse point clouds and pre-align them with the ground truth based on the Earth Mover's Distance (EMD) (Chen and et al. 2020). Similar to most patch-based point cloud upsampling (He and et al. 2023; Qu and et al. 2024; Li et al. 2019; Qian and et al. 2021), we extract the paired sparse and dense patches from the original point clouds to optimize the flow matching model. During the inference, we can take the whole sparse point cloud as input for upsampling, or we can apply patch based upsampling and then assemble all upsampled patches to form the complete target data.

Point cloud flow matching. Given the sparse point cloud $x_0 \sim \mathcal{X}_{\text{sparse}} \in \mathbb{R}^{M\times 3}$ and dense point clouds $x_1 \sim \mathcal{X}_{\text{dense}} \in \mathbb{R}^{N\times 3}$. Our goal is to learn a flow matching model by finding the optimal transport between $\mathcal{X}_{\text{sparse}}$ and $\mathcal{X}_{\text{dense}}$. Since the

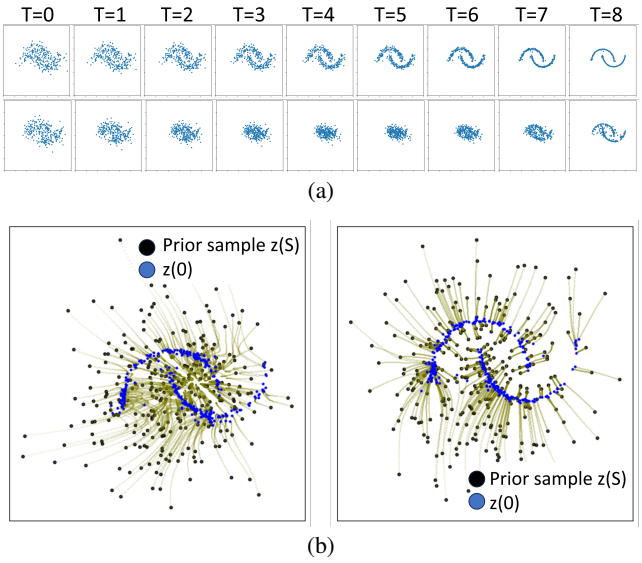

Figure 3: A example of matching 2D moon-shaped point clouds from Gaussian noise. (a) Without pre-alignment, the model first transforms the source point clouds to a set of dispersed clusters and then gradually matches to the target point clouds. With pre-alignment, the model exhibits a more efficient and consistent transformation. (b) The matching trajectories with or without applying pre-alignment.

sparse point cloud is a partial view of dense point clouds but shares the same underlying 3D structure, we initially densify the sparse point cloud via a midpoint interpolation (He and et al. 2023) method. Given the sparse point clouds x_0, its densified version $\tilde{x}_0 = \text{mid}(x_0, \eta)$ can be defined as:

$$\tilde{x}_0 = \frac{1}{2} \cdot [\text{R}_\gamma(x_0) + \text{FPS}(x_0, \gamma)] + \eta n, \quad \text{where } n \sim \mathcal{N}(0, I) \quad (5)$$

where $\text{R}_\gamma(\cdot)$ means repeating points by γ times and FPS denotes the Furthest Point Sampling (Yifan and et al. 2019). Without losing generalization, we also add a small portion of Gaussian noise with noise level η to the sparse point cloud to simulate real-world noisy point clouds. The objective of learning a point cloud upsampling model can then be interpreted as learning the velocity field from two distributions. We parameterize the velocity field by a network $\nu_\theta(x_t, t)$ with learnable parameter θ, and learn the network by minimizing Equation.3. The interpolant x_t are sampled from the matching path based on the timestep t:

$$x_t = (1 - t)\tilde{x}_0 + t x_1. \quad (6)$$

This framework enables effective learning of a flow matching model to upsample sparse point clouds by aligning them with dense point clouds through optimal transport.

Pre-alignment. Since the sparse point clouds are randomly sampled from their dense counterparts with additional noise, both $\tilde{x}_0$ and x_1 have an unordered format, which leads to a misalignment between the sparse and dense point clouds. The residual signal $x_1 - \tilde{x}_0$ between two distributions varies significantly, leading to ambiguous learning by $\nu_\theta(x_t, t)$. As shown in Figure 1b, the model finds it easier to spread the points uniformly across space. This causes a set of diffused ball-shaped clusters since the model will first solve these irregularities between these unpaired points by dispersing the points to a more tractable distribution, rather than approaching the target distribution.

We performed a toy experiment on denoising 2D moon-shape point clouds to show why pre-alignment matters. Figure 3a shows that without pre-alignment, the noisy point cloud turns into random noise before slowly rebuilding its shape, while pre-aligned point clouds move straight to the target shape and converge faster. Figure 3b indicates that the paths learned from unaligned data get tangled, unlike the straighter paths from aligned data. Importantly, pre-alignment is used only during training, which helps make our model efficient without requiring an extra computational budget.

To this end, we align the point clouds by finding an optimum assignment $\phi^* = \phi(\tilde{x}_0, x_1)$ based on their Earth Mover's Distance (EMD) (Chen and et al. 2020). For each point in the $\mathbf{x}_1$ and its nearest neighbor in $\mathbf{x}_0$, the assignment function can be defined as:

$$\phi^* = \underset{\phi}{\text{argmin}} \, \Phi \sum_{i=1}^{N} \|x_1^{\phi(i)} - \tilde{x}_0^i\|_2, \quad (7)$$

where $\Phi = 1, ..., N \to 1, ..., N$ is the set of possible bijective assignments between points in x_1 and $\tilde{x}_0$. During the training stage, we first apply ϕ^* to permute the dense point clouds, aligning them with the sparse point clouds. This alignment ensures that the model learns from a consistent pairing of sparse and dense point clouds. Next, we sample the interpolant x_t from the paired data. As shown in Figure 1, our approach demonstrates faster convergence when pre-alignment is applied, and high-fidelity dense point clouds can be generated in just a few sampling steps.

Training and sampling We adopt a cosine annealing schedule (Nichol and Dhariwal 2021) for timestep sampling, where t is drawn as $t = 1 - \cos(s\pi/2)$ with $s \sim \mathcal{U}[0, 1]$. This schedule emphasizes earlier timesteps during training. A comparison of using different sampling strategies is included in the supplementary material.

Model architecture We follow previous works (Luo and Hu 2021; Qu and et al. 2024; Vogel and et al. 2024) and adopt a PointNet++-based (Qi and et al. 2017) architecture with a U-Net structure composed of set abstraction (SA) and feature propagation (FP) blocks to learn multiscale point features. To enhance global feature interaction, we insert vector attention blocks (Guo 2021; Wu and et al. 2022) at the final SA layer. The network is conditioned on the timestep t using sinusoidal positional embeddings $\gamma(t) \in \mathbb{R}^d$. These embeddings are injected into both SA and FP blocks through scale-and-shift modulation. Specifically, for a given feature vector f, the time-conditioned output is computed as:

$$\text{Mod}(f, \gamma(t)) = \mathbf{s}_t \odot f + \mathbf{b}_t, \quad (8)$$

where $(\mathbf{s}_t, \mathbf{b}_t) = \text{MLP}(\gamma(t))$ are scale and shift vectors generated from the time embedding, and $\odot$ denotes element-wise multiplication. This mechanism allows the network to adapt its feature extraction dynamically over time.

Upscaling factors		4×			16×		
Dataset	Method	CD	HD	P2F	CD	HD	P2F
PUGAN	PUBP	1.649	1.476	5.997	0.982	2.071	7.496
	PUGCN	2.774	3.831	9.508	1.102	1.785	7.125
	RepKPU	1.067	1.139	1.974	0.384	1.245	2.151
	PUGAN	1.541	1.391	5.420	0.869	1.746	6.757
	PUDM	1.221	1.174	3.132	0.533	1.185	3.589
	Grad-PU	1.132	1.186	1.957	0.415	1.142	2.185
	Ours	**1.049**	**0.876**	**1.864**	**0.353**	**0.844**	**2.103**
PU1K	PUBP	0.694	0.593	2.206	0.343	0.712	2.958
	PUGCN	1.241	1.504	5.115	2.393	4.214	9.410
	RepKPU	0.566	0.619	1.464	0.290	0.592	**1.821**
	PUGAN	0.682	0.632	2.792	0.361	0.773	3.538
	PUDM	0.706	0.605	2.891	0.421	0.602	3.370
	Grad-PU	0.626	0.583	1.510	0.316	**0.552**	1.890
	Ours	**0.545**	**0.556**	**1.770**	**0.220**	0.578	1.983

Table 1: Overall comparison with state-of-the-art methods. We report CD (10^{-4}), HD (10^{-3}) and P2F (10^{-3}) on PUGAN and PU1K datasets.

Method	PUGAN			PU1K			Runtime (s) /
	CD	HD	P2F	CD	HD	P2F	# Steps
PUBP	1.649	1.476	5.997	0.694	0.593	2.206	0.204
PUGCN	2.774	3.831	9.508	1.241	1.504	5.115	0.181
PUGAN	1.541	1.391	1.974	0.682	0.632	2.792	0.505
Grad-PU	1.132	1.186	1.957	0.626	0.583	1.510	0.568/2
RepKPU	1.067	1.139	1.974	0.566	0.619	**1.464**	1.021
DDPM	1.239	1.558	3.207	0.618	0.739	2.616	7.83/100
InDI	1.542	1.529	6.033	0.737	0.730	3.903	0.82/10
PUDM	1.221	1.174	3.132	0.706	0.605	2.891	1.83/30
Ours	**1.049**	**0.876**	**1.864**	**0.545**	**0.556**	1.770	**0.71/5**

Table 2: Comparison with state-of-the-art methods. Performance, runtime and sampling steps on PUGAN and PU1K datasets are reported.

Experiments

Experimental Details

Dataset. We use two public datasets, PUGAN (Li et al. 2019) and PU1K (Qian and et al. 2021), and follow the same procedure in (Yifan and et al. 2019) to extract paired sparse and dense point clouds. Given the 3D meshes, we first use Poisson disk sampling to generate uniform patches as ground truth, each patch contains 1024 points. Then we randomly sample 256 points from each patch to obtain the sparse input data. For testing, we obtain 27 and 127 point clouds from PUGAN and PU1K, respectively.

Evaluation metrics. We use the Chamfer Distance (CD), Hausdorff Distance (HD) and Point-to-Surface (P2F) as the evaluation metrics in our experiments.

Comparison with State-of-the-Art

We quantitatively evaluate our method against several point cloud upsampling approaches, including PUBP (Liu and et al. 2023), PUGCN (Qian and et al. 2021), PUGAN (Li et al. 2019), PUDM (Qu and et al. 2024), Grad-PU (He and et al. 2023), and RepKPU (Rong and et al. 2024). For fair comparison, we adopt the training settings and pretrained models from their official implementations. We evaluate 4× and 16× upsampling (twice 4× upsampling) on randomly sampled sparse point clouds. As shown in Table 1, our method achieves the best CD, HD, and P2F scores, indicating superior fidelity compared to both diffusion-based and other learning-based methods. Figure 4 visualizes the 4× upsampling results. Highlighted regions show that our method produces evenly distributed points with fewer outliers. Surface reconstruction (Hoppe and et al. 1992; Bernardini and et al. 1999) further demonstrates that our results are smoother and more faithful to the underlying geometry.

We also compare our model with other iterative inversion approaches, such as DDPM (Ho and et al. 2020), and InDI (Delbracio and Milanfar 2023). Table 2 shows that ours reduces the CD distance by approximately 0.1∼0.2 points and HD distance by approximately 0.1∼0.2 points. In addition, we report the optimal sampling steps and the runtime according to their official results. Our model requires fewer sampling steps, leading to faster inference.

Method	Grad-PU		PUDM		Ours	
	CD	HD	CD	HD	CD	HD
5×	0.952	4.841	1.054	5.091	**0.921**	**4.763**
7×	0.752	4.523	0.850	4.537	**0.708**	**4.363**
8×	0.686	3.880	0.784	4.054	**0.638**	**3.947**
12×	0.515	3.628	0.617	**3.371**	**0.455**	3.380
20×	0.452	3.238	0.473	2.695	**0.344**	**2.666**
24×	0.412	3.108	0.444	2.691	**0.314**	**2.690**
32×	0.365	2.667	0.404	**2.322**	**0.273**	2.517

Table 3: Arbitrary point clouds upsampling on PUGAN.

Noise level	η =0.01			η =0.02		
Method	CD	HD	P2F	CD	HD	P2F
PUDM	1.706	**1.803**	6.025	3.100	3.622	1.150
RepKPU	1.696	2.189	6.989	3.974	4.827	1.615
Grad-PU	1.795	1.965	6.591	3.588	4.248	1.310
Ours	**1.496**	1.879	**5.887**	**2.404**	**2.794**	**1.055**

Table 4: 4× point cloud upsampling on noisy PUGAN.

Arbitrary Upsampling and Denoising

Our proposed method can produce multi-scale point cloud upsampling as others (He and et al. 2023; Qu and et al. 2024) by repeatedly applying 4× upsampling and FPS downsampling. To show the versatility of ours, we compare ours with Grad-PU and PUDM on 5× to 32× upsampling and report the results in Table 3. We can see that ours steadily outperforms the other two methods in CD, HD, and P2F. We also show *Chair* example from PUGAN in Figure 5. We observe that PUDM produces holes on the surface of the chair, while ours can produce evenly distributed points without deviating from the underlying surface.

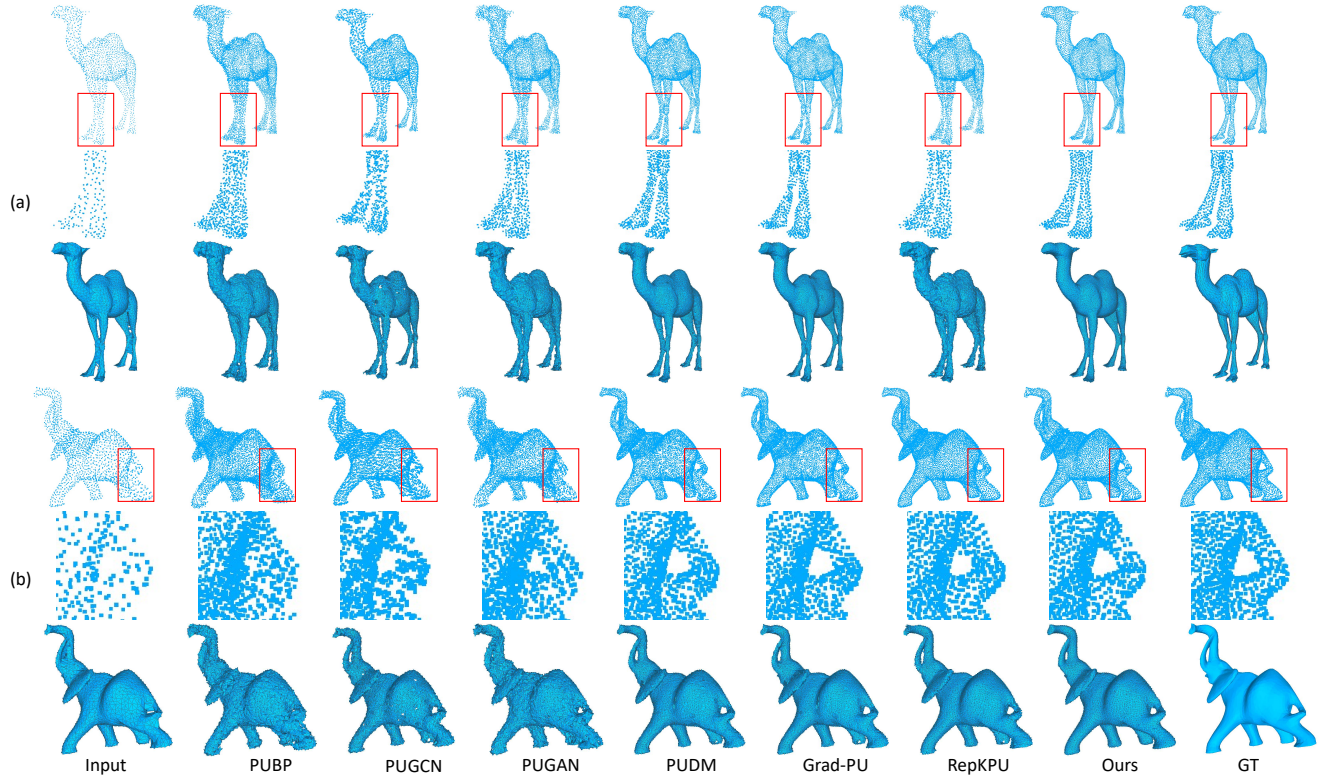

Figure 4: Visual comparison of different methods on 4× upsampling. Our method generates evenly distributed points and smooth surfaces, while others exhibit obvious holes (see the elephant and camel legs) and noisy surfaces.

Alignment Method	PUGAN			PU1K		
	CD	HD	P2F	CD	HD	P2F
NoAlign	5.167	4.417	2.128	3.025	4.205	8.56
CD	1.143	1.214	1.922	0.662	0.815	1.815
EMD (ours)	**1.049**	0.876	**1.864**	**0.545**	**0.556**	**1.770**

Table 5: Ablation on different alignment strategies. CD, HD, and P2F scores on PUGAN and PU1K datasets.

Time Scheduler	PUGAN			PU1K		
	CD	HD	P2F	CD	HD	P2F
Uniform	1.184	1.225	2.548	0.624	0.598	1.924
Cosine (ours)	**1.049**	0.876	**1.864**	**0.545**	**0.556**	**1.770**

Table 6: Ablation on time scheduling strategies. Comparison between Uniform and CosineAnnealing schedulers on PUGAN and PU1K datasets.

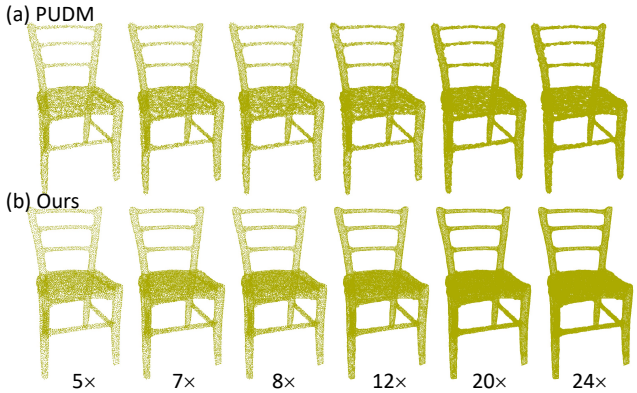

Figure 5: Arbitrary-scale point cloud upsampling.

Ablation Studies

We first evaluate the effectiveness of the proposed EMD-based pre-alignment method. As shown in Table 5, applying EMD significantly improves upsampling performance across all metrics on both PUGAN and PU1K datasets. In contrast, removing alignment or using Chamfer Distance (CD) for alignment leads to noticeably worse results. These results indicate that providing accurate initial point correspondences is crucial for flow matching. Compared to CD, EMD better preserves the local geometric structure, offering more reliable coarse matches that guide the subsequent restoration process more effectively. Furthermore, we utilize auction-based EMD approximation (Chen and et al. 2020). Given a batch input of $32 \times 3 \times 1024$, the EMD process costs 0.0021s, and the model forward pass is 0.044s, which means that the EMD only introduces 5% computation overhead.

We also compare different time schedulers in Table 6. Cosine annealing approach outperforms the uniform scheduler

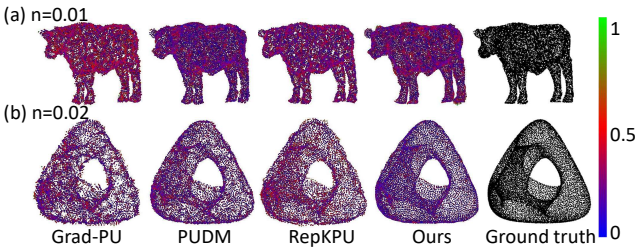

Figure 6: Visualization of noisy point cloud upsampling. We calculate the P2F distance as the color feature to the point cloud for visualization. Ours are visually better than baselines, without showing holes in (a) and noisy outliers in (b).

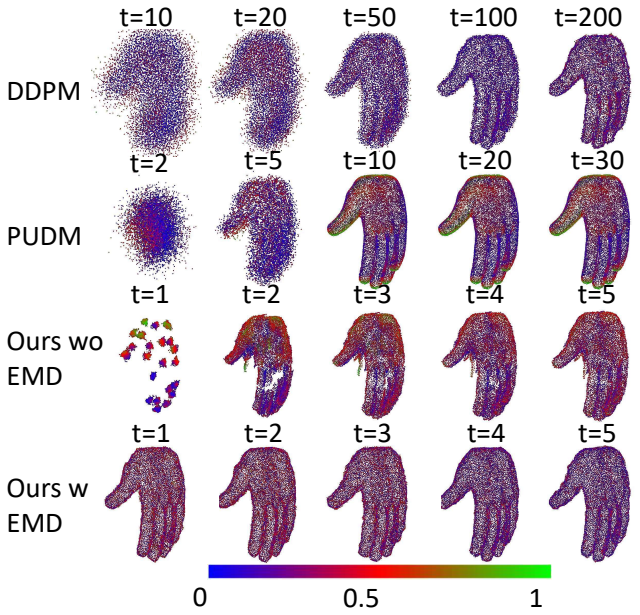

Figure 7: Visualization at different sampling steps. Colors indicate the P2F distances. Ours quickly learns the optimal upsampling point clouds, while others take longer sampling steps (> 10).

in CD, HD, and P2F. This advantage stems from its inherent ability to facilitate a coarse-to-fine refinement process during training or inference. Specifically, it allocates larger step sizes in the early phases, allowing for rapid exploration and broad adjustments to the model's outputs, while progressively reducing the step sizes in later stages to enable more precise, finer-grained updates.

In Figure 7, we present intermediate samples generated at various time steps. Our method demonstrates rapid convergence toward the ground truth, yielding outputs with markedly fewer errors. In scenarios without pre-alignment, the process initially collapse into dispersed clusters before progressively refining to align with the ground truth.

Robustness and Downstream Applications

To demonstrate the robustness of our method to noise, we add Gaussian noise with different levels (η) and apply $4\times$ upsampling on PUGAN data. From Table 4, it can be seen that ours consistently performs better than PUDM, RepKPU and Grad-PU at different noise levels, with approximately 0.3, 0.7 and 1.1 drops in terms of CD, HD and P2F, repectively. Figure 6 visually demonstrates that our method maintains structural integrity and produces denser, more accurate point clouds, even under high noise conditions.

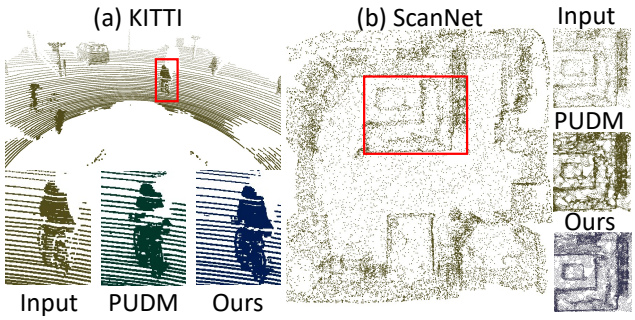

Figure 8: Visulization of real point cloud upsampling. We use two examples from ScanNet and KITTI to apply $4\times$ upsampling. Compared to PUDM, ours does not show inconsistent patterns on the bicyclist or noisy point distribution in the living room.

We also evaluate the upsampling performance of our method on real-world data. We test ours and PUDM on ScanNet (Dai et al. 2017) and KITTI (Geiger and et al. 2013) datasets. We visualize the upsampled point clouds in Figure 8. Our method demonstrates superior performance over PUDM in multiple scenarios. In (a) KITTI, our approach captures bicyclists with high fidelity, producing continuous depth maps without gaps in human body structures. In (b) ScanNet, our approach generates uniformly distributed dense point clouds, revealing consistent indoor 3D layouts.

Conclusion

In this work, we present Point cloud Upsampling via Flow Matching (PUFM), a novel and efficient framework designed to learn the direct optimal transport mapping between sparse input point clouds and their dense counterparts. Instead of directly minimizing the l_2 positional loss between two distributions, which often struggle with the unordered and irregular nature of point clouds, our method employs optimization with EMD to pre-align sparse and dense point clouds, enabling the learning of a more consistent trajectory in the flow matching objective. Comprehensive experiments on both synthetic and real-world datasets demonstrate that PUFM consistently outperforms state-of-the-art methods in terms of accuracy and robustness. Additionally, PUFM achieves high efficiency compared to diffusion-based approaches and demonstrates strong resilience to noise and scalability across different input densities.

Acknowledgments

Chenhang He acknowledges support from the National Natural Science Foundation of China (No.62406268).

References

Bernardini, F.; and et al. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Vis. Comput. Graph.*, 5(4): 349–359.

Charles, R.; Su, H.; Kaichun, M.; and Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 77–85. Los Alamitos, CA, USA.

Chen, H.; and et al. 2023. Deep Point Set Resampling via Gradient Fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3): 2913–2930.

Chen, Y.; and et al. 2020. PointMixup: Augmentation for Point Clouds. In *Eur. Conf. Comput. Vis.*

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*.

Delbracio, M.; and Milanfar, P. 2023. Inversion by Direct Iteration: An Alternative to Denoising Diffusion for Image Restoration. *Trans. Mac. Lear. Res.*

Du, Y.; Zhao, Z.; Su, S.; Golluri, S.; Zheng, H.; Yao, R.; and Wang, C. 2025. SuperPC: A Single Diffusion Model for Point Cloud Completion, Upsampling, Denoising, and Colorization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 16953–16964.

Feng, W.; and et al. 2022. Neural Points: Point Cloud Representation with Neural Fields for Arbitrary Upsampling. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Geiger, A.; and et al. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*.

Guo, e. a., MH. 2021. PCT: Point cloud transformer. *Comp. Visual Media*, 10(7): 187–199.

Hanocka, R.; and et al. 2020. Point2Mesh: A Self-Prior for Deformable Meshes. *ACM Trans. Graph.*, 39(4).

He, Y.; and et al. 2023. Grad-PU: Arbitrary-Scale Point Cloud Upsampling via Gradient Descent with Learned Distance Functions. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5354–5363.

Ho, J.; and et al. 2020. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.* ISBN 9781713829546.

Hoppe, H.; and et al. 1992. Surface reconstruction from unorganized points. In *Proceed. Computer Graphics and Interactive Techniques*, 71–78.

Kerbl, B.; and et al. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Trans. Graph.*, 42(4).

Kumbar, A.; and et al. 2023. TP-NoDe: Topology-aware Progressive Noising and Denoising of Point Clouds towards Upsampling. In *Int. Conf. Comput. Vis.*, 2264–2274.

L., Y.; and et al. 2007. Parameterization-Free Projection for Geometry Reconstruction. In *ACM SIGGRAPH*, 22–es. New York, NY, USA.

Lemke, O.; and et al. 2024. Spot-Compose: A Framework for Open-Vocabulary Object Retrieval and Drawer Manipulation in Point Clouds. In *Int. Conf. on Robotics and Automation*.

Li, J.; Pang, B.; and Wang, P.-S. 2024. Joint Point Cloud Upsampling and Cleaning with Octree-based CNNs. *arXiv preprint arXiv:2410.17001*.

Li, R.; and et al. 2021. Point Cloud Upsampling via Disentangled Refinement. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Li, R.; Li, X.; Fu, C.; Cohen-Or, D.; and Heng, P. 2019. PU-GAN: A Point Cloud Upsampling Adversarial Network. *Int. Conf. Comput. Vis.*, 7202–7211.

Lipman, Y.; and et al. 2023. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*.

Liu, Z.-S.; and et al. 2023. Arbitrary Point Cloud Upsampling Via Dual Back-Projection Network. In *IEEE Int. Conf. Image Process.*, 1470–1474.

Lu, X.; and et al. 2022. Low Rank Matrix Approximation for 3D Geometry Filtering. *IEEE Trans. Vis. Comput. Graph.*, 28(4): 1835–1847.

Luo, S.; and Hu, W. 2021. Diffusion Probabilistic Models for 3D Point Cloud Generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Ma, X.; and et al. 2022. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. *Int. Conf. Learn. Represent.*

Mao, A.; Du, Z.; Hou, J.; Duan, Y.; Liu, Y.-J.; and He, Y. 2023. PU-Flow: A Point Cloud Upsampling Network With Normalizing Flows. *IEEE Trans. Vis. Comput. Graph.*, 29(12): 4964–4977.

Metzer, G.; and et al. 2021. Self-Sampling for Neural Point Cloud Consolidation. *ACM Transactions on Graphics*, 40(5).

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Park, C.; and et al. 2022. Fast Point Transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 16949–16958.

Qi, C. R.; and et al. 2017. PointNet++: deep hierarchical feature learning on point sets in a metric space. In *Adv. Neural Inform. Process. Syst.*, NIPS'17, 5105–5114.

Qian, G.; and et al. 2021. PU-GCN: Point Cloud Upsampling Using Graph Convolutional Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 11683–11692.

Qian, G.; and et al. 2024. PointNeXt: revisiting PointNet++ with improved training and scaling strategies. In *Adv. Neural Inform. Process. Syst.*, NIPS '22.

Qian, Y.; and et al. 2020. PUGeo-Net: A Geometry-centric Network for 3D Point Cloud Upsampling. *arXiv*, abs/2002.10277.

Qu, W.; and et al. 2024. A Conditional Denoising Diffusion Probabilistic Model for Point Cloud Upsampling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 20786–20795.

Rong, Y.; and et al. 2024. RepKPU: Point Cloud Upsampling with Kernel Point Representation and Deformation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 21050–21060.

Singh, G.; and et al. 2007. Guest Editors' Introduction: Special Section on ACM VRST 2005. *IEEE Trans. Vis. Comput. Graph.*, 9(01): 3–4.

Thomas, H.; and et al. 2019. KPConv: Flexible and Deformable Convolution for Point Clouds. *Int. Conf. Comput. Vis.*

Vogel, M.; and et al. 2024. P2P-Bridge: Diffusion Bridges for 3D Point Cloud Denoising. In *Eur. Conf. Comput. Vis.*

Wang, Y.; and et al. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.*, 38(5).

Wenbo, Z.; and et al. 2022. Self-Supervised Arbitrary-Scale Point Clouds Upsampling via Implicit Neural Representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Wu, X.; and et al. 2022. Point transformer V2: Grouped Vector Attention and Partition-based Pooling. In *Adv. Neural Inform. Process. Syst.*

Wu, X.; and et al. 2024. Point Transformer V3: Simpler, Faster, Stronger. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Wu, X.; Lao, Y.; Jiang, L.; Liu, X.; and Zhao, H. 2022. Point transformer V2: Grouped Vector Attention and Partition-based Pooling. In *Adv. Neural Inform. Process. Syst.*

Yang, Z.; and et al. 2024. Visual Point Cloud Forecasting enables Scalable Autonomous Driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*

Yifan, W.; and et al. 2019. Patch-Based Progressive 3D Point Set Upsampling. *IEEE Conf. Comput. Vis. Pattern Recog.*, 5951–5960.

Yu, L.; and et al. 2018. PU-Net: Point Cloud Upsampling Network. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2790–2799.

Zhang, K.; and et al. 2019. Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features. *ArXiv*, abs/1904.10014.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Int. Conf. Comput. Vis.*, 16259–16268.