LiFlow: Flow Matching for 3D LiDAR Scene Completion

Andrea Matteazzi and Dietmar Tutsch

University of Wuppertal, Germany
{matteazzi,tutsch}@uni-wuppertal.de

Abstract. In autonomous driving scenarios, the collected LiDAR point clouds can be affected by occlusion and long-range sparsity, limiting the perception of autonomous driving systems. Scene completion methods can infer the missing parts of incomplete single LiDAR scans. Recent methods adopt point-level denoising diffusion probabilistic models that rely on approximations to handle scene-scale data, leading to a mismatch between training and inference initial distributions. We propose LiFlow, the first point-level flow matching method for 3D LiDAR scene completion. LiFlow improves upon existing diffusion-based methods by directly aligning single LiDAR scans to complete scenes, ensuring consistent initial distributions between training and inference. LiFlow introduces nearest neighbor flow matching and Chamfer distance matching to enhance both local structure and global coverage in the alignment of point clouds. Compared to existing diffusion-based methods, LiFlow reduces the number of inference steps, enabling more efficient scene completion while maintaining stability and high generation quality. Extensive experiments demonstrate that LiFlow achieves state-of-the-art performance across multiple metrics. Code is available at `https://github.com/matteandre/LiFlow`.

Keywords: LiDAR · Scene completion · Flow matching

1 Introduction

LiDAR scenes constitute a fundamental source of data for perception modules, enabling autonomous driving systems to perceive and understand the surrounding environment. LiDAR sensors provide 3D information of the vehicle surroundings in the form of 3D point clouds. The reliability of LiDAR-based perception modules therefore depends on the accuracy and completeness of the 3D point clouds. Despite the accuracy of LiDAR sensors, in autonomous driving scenarios, the collected point clouds are affected by occlusion and long-range sparsity. The resulting gaps in the collected point clouds can limit perception systems from perceiving objects and structures in the environment. Scene completion methods can infer the missing parts of incomplete 3D scenes. Providing a dense and more complete scene representation of non-observed regions can add valuable information to autonomous driving systems, helping to improve different tasks such as object detection [36], semantic segmentation [22] or navigation [25].

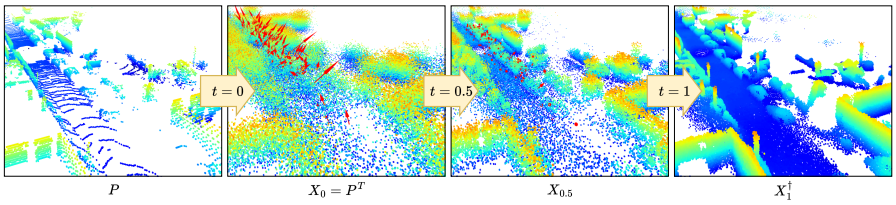

Fig. 1: Illustration of our flow matching method. Starting from a single LiDAR Scan P from the validation set of the SemanticKITTI dataset (sequence 08), we compute the initial point cloud P^T. LiFlow generates a flow of point clouds with Euler's method [14] by predicting time-dependent vector fields $u^*_\theta(t, X_t, P)$ in continuous time $t\colon 0 \to 1$, achieving a 3D complete scene. The arrows indicate a scaled version of $u^*_\theta(t, X_t, P)$ for some points labeled in P as car. Colors depict point height. †: with refinement network [24].

Recent methods adopt point-level Denoising Diffusion Probabilistic Models (DDPMs) [5] that rely on various approximations to handle scene-scale data, such as altering the forward diffusion process to constrain noise locally [24] or truncating the starting point of the reverse denoising process [21]. While effective, these methods introduce a mismatch between the initial distributions used during training and inference.

We propose LiFlow, the first Flow Matching (FM) [13] method for 3D LiDAR scene completion at the point level (see Fig. 1). To address the mismatch between initial distributions, we propose an FM formulation that includes Nearest Neighbor Flow Matching (NFM) and Chamfer Distance Matching (CDM). NFM enables the definition of a data-dependent coupling that directly aligns single LiDAR scans to complete scenes, ensuring consistent initial point cloud distributions between training and inference. CDM improves the alignment of the point clouds, promoting better coverage of the complete scene. Our FM formulation enables the application of Optimal Transport (OT) [33] at scene scale, allowing LiFlow to reduce the number of inference steps and facilitating more efficient scene completion while maintaining stability and high generation quality. We compare LiFlow with different scene completion methods and conduct extensive experiments to validate our proposed flow matching method.

In summary, our key contributions are:

- We propose the first flow matching method for 3D LiDAR scene completion that operates at the point level.
- We formulate nearest neighbor flow matching and Chamfer distance matching to directly align single LiDAR scans to complete scenes, ensuring consistent initial distributions between training and inference.
- Our flow matching formulation enables the application of optimal transport at scene scale, reducing the number of inference steps while maintaining stability and high generation quality compared to existing diffusion-based methods.

– Our method achieves state-of-the-art results compared to previous diffusion and non-diffusion methods.

2 Related Work

2.1 Scene Completion

Scene completion is the task of inferring the missing parts of an incomplete 3D scene provided by a sensor. Given the LiDAR data sparsity, providing a dense and more complete scene representation of non-observed regions is fundamental to adding valuable information to perception modules. Early works [4, 8, 19, 20] propose scene completion as a prediction of dense depth images extracted from RGB images and fused with LiDAR point clouds. Recent approaches address the scene completion task in 3D space for denser completions. Some methods [12,35] leverage Signed Distance Fields (SDF) to represent 3D LiDAR scenes as voxel grids, where each voxel stores the distance to the nearest surface, and they train a 3D convolutional network to predict such SDF volumes from LiDAR scenes. In addition, further methods operating at the voxel level [27,37,39] aim to solve semantic scene completion by leveraging and inferring semantic labels for each voxel. The scene completion accuracy of methods operating at the voxel level is intrinsic to the voxel grid resolution. DDPMs [5] show promising results in the generation of 3D point clouds of single-object shapes [11, 18, 40] and more recently in the completion of 3D LiDAR scenes at the point level [21, 24].

2.2 Diffusion Models for 3D Scene Completion

LiDiff [24] and LiDPM [21] address 3D LiDAR scene completion by adopting point-level DDPMs that rely on various approximations to handle scene-scale data. LiDiff defines the forward process during training by adding independent Gaussian noise to each point of a complete 3D scene, producing strictly local perturbations. As a result, the initial distribution remains dependent on the complete scene, and inference cannot start from Gaussian samples. Instead, a surrogate initial distribution is created by applying the same noise procedure to the observed single LiDAR scan, since the complete scene is not accessible at inference. LiDiff further proposes a refinement network that upsamples and refines the complete scene generated by the diffusion process. For each generated point, the network predicts κ offsets that are applied point-wise to the corresponding point, resulting in κ refined points. LiDPM follows the standard diffusion formulation but starts the reverse denoising process from an intermediate diffusion step instead of Gaussian noise. Although this improves empirical performance compared to LiDiff, the initial distribution still depends on the complete scene and therefore requires a surrogate initial distribution constructed from the observed single LiDAR scan at inference. While starting from Gaussian noise, as in vanilla DDPMs, would avoid this mismatch, it has been observed to degrade performance in scene-scale settings [21, 24].

As a result, both approaches depend on approximations of the initial distribution at inference, introducing a distribution mismatch between training and inference. In contrast, our method uses the single LiDAR scan, with local noise applied, for both training and inference, thereby avoiding this mismatch and producing more consistent 3D scene completion.

2.3 Flow Matching for 3D Data

While DDPMs represent a discrete time formulation of diffusion and denoising, Song *et al.* [31] prove its equivalent formulation in continuous time. In continuous time, denoising steps are equivalent to solving a Stochastic Differential Equation (SDE) [16]. The more general FM [13] paradigm extends the continuous diffusion formulation with an equivalent Ordinary Differential Equation (ODE) formulation not restricted to Gaussian distributions. The FM formulation allows for mapping between initial and target distributions without assuming a Gaussian relationship between them, and enables faster convergence by not introducing stochasticity into the matching procedure [29]. FM demonstrates significant improvements over DDPMs in different image generation datasets [13, 28].

Recent works [10, 15] leverage FM for the generation of 3D point clouds of single-object shapes. Liu *et al.* [15] propose to upsample 3D point clouds by introducing a pre-alignment method based on Earth Mover's Distance (EMD) optimization to ensure coherent interpolation between sparse and dense point clouds. To the best of our knowledge, prior work has focused on FM for single-object 3D point clouds, leaving its application to scene-scale 3D LiDAR point clouds underexplored. In contrast, we are the first to apply FM to scene-scale, multi-object 3D LiDAR scene completion at the point level.

3 Preliminary

3.1 Flow Matching

FM [13] is a paradigm for generative modeling that regresses vector fields to generate target probability density paths. Let $x \in \mathbb{R}^d$ be the data points. We can define a probability density path $p_t \colon [0,1] \times \mathbb{R}^d \to \mathbb{R}_{>0}$, as a time-dependent probability density function, *i.e.*, $\int p_t(x)dx = 1$, and a time-dependent vector field $v_t \colon [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. A vector field v_t can be used to construct a flow $\phi_t \colon [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, defined as the solution of the ordinary ODE:

$$\begin{aligned} \phi_0(x) &= x, \\ \frac{\mathrm{d}\phi_t(x)}{\mathrm{d}t} &= v_t(\phi_t(x)). \end{aligned} \tag{1}$$

A vector field v_t induces a probability density path p_t if its flow ϕ_t satisfies the push-forward equation:

$$p_t(x) = [\phi_t]_* p_0(x) = p_0(\phi_t^{-1}(x)) \det \left[\frac{\partial \phi_t^{-1}(x)}{\partial x} \right]. \tag{2}$$

Let $x_1 \sim q$ be a data sample drawn from the unknown data distribution q. We assume access only to samples from q, but not to the density function itself. Let p_t be a probability density path such that $p_0 = p$ is a simple initial distribution and p_1 is approximately equal in distribution to q. The FM objective is designed to match this target probability path, allowing us to flow from p_0 to p_1. Given a target vector field $v_t(x)$ generating the target probability density path p_t, *i.e.*, $x_t = \phi_t(x_0) \sim p_t$, $x_0 \sim p_0$, we can regress a vector field $u_\theta(t, x)$ parameterized by a neural network with learnable parameters θ using the FM objective as:

$$\mathcal{L}_{FM} = \mathbb{E}_{t,x_0}\left[\left\|u_\theta(t, \phi_t(x_0)) - v_t(\phi_t(x_0))\right\|^2\right]. \tag{3}$$

Since we generally do not have access to a closed form of v_t, we can acquire the same gradients and therefore efficiently regress the neural network using the Conditional Flow Matching (CFM) objective:

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,x_0,x_1}\left[\left\|u_\theta(t, \phi_t(x_0|x_1)) - v_t(x_0|x_1)\right\|^2\right], \tag{4}$$

with $\phi_t(x_0|x_1)$ conditional flow and $v_t(x_0|x_1)$ conditional vector field.

3.2 Data-Dependent Couplings

The data-dependent couplings [14,29] formulation allows us to create a probability path between an initial signal and target data within the FM objective. Let x_1 be target data and x_0 be an initial signal obtained from x_1. We can smooth around the data samples within a minimal variance to acquire the corresponding data distribution $\mathcal{N}(x_0, \sigma^2_{min} I)$ and $\mathcal{N}(x_1, \sigma^2_{min} I)$. The flow can be associated with Optimal Transport (OT) and defined by the equations:

$$\begin{aligned} \phi^C_t(x_0|x_1) &= tx_1 + (1-t)x_0, \\ v^C_t(x_0|x_1) &= x_1 - x_0. \end{aligned} \tag{5}$$

The corresponding conditional probability path is:

$$p_t(x_t|x_1) = \mathcal{N}(x_t|tx_1 + (1-t)x_0, \sigma^2_{min} I). \tag{6}$$

The resulting Data-Dependent Coupling Flow Matching (DFM) loss becomes:

$$\mathcal{L}_{DFM} = \mathbb{E}_{t,x_0,x_1}\left[\left\|u_\theta(t, \phi^C_t(x_0|x_1)) - (x_1 - x_0)\right\|^2\right]. \tag{7}$$

4 Method

We present LiFlow as the first FM [13] method for 3D LiDAR scene completion. LiFlow aims to solve the mismatch in the initial point cloud distribution between training and inference analyzed previously. Thanks to the flexibility of FM, we can use the same initial point cloud distribution for both training and inference.

4.1 Flow Matching for 3D Scene Completion

Let $G \in \mathbb{R}^{M\times 3}$ be a 3D LiDAR complete scene and $P \in \mathbb{R}^{N\times 3}$ a single LiDAR scan. Similarly to LiDiff [24], we generate $P^* \in \mathbb{R}^{M\times 3}$ from P by concatenating its points K times such that $M = KN$. We then compute the initial noisy point cloud P^T from P^* as a noise offset added locally to each point $\mathbf{p}^* \in P^*$:

$$\mathbf{p}^T = \mathbf{p}^* + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{8}$$

Following the data-dependent couplings [14, 29], we set the target data $x_1 = G$ and the initial signal $x_0 = P^T$. We further condition FM with classifier-free guidance [6] with condition c associated with the single LiDAR scan P, *i.e.*, $\tilde{c} \sim \mathbb{B}(p)$ as a Bernoulli distribution of outcomes $\{\emptyset, P\}$ with probability p that $\emptyset$ occurs.

This formulation enables training and inference from the same initial distribution, addressing the mismatch issue of existing diffusion-based methods [21, 24]. However, a point-wise correspondence between x_0 and x_1 is not yet established, as the points originate from different point clouds.

4.2 Nearest Neighbor Flow Matching

To satisfy the data-dependent coupling in Eq. (5), we introduce point-wise Nearest Neighbor Flow Matching (NFM) to align a noisy point cloud x_0 with the target x_1. NFM establishes point-wise correspondences by assigning each point in x_0 to its nearest neighbor in x_1. Unlike LiDiff, which assumes Gaussian point-wise correspondences a priori, NFM adapts a posteriori to the initial distribution P^T. This enables the definition of a data-dependent coupling that directly aligns single LiDAR scans to complete scenes, ensuring consistent initial distributions between training and inference. Formally, the Nearest Neighbor (NN) of each point in x_0 is defined as the point in x_1 minimizing the $\mathcal{L}_2$ distance over 3D coordinates:

$$NN(P, \hat{P}) = \left(\underset{\hat{\mathbf{p}}\in\hat{P}}{\arg\min} \|\mathbf{p} - \hat{\mathbf{p}}\| \right)_{\mathbf{p}\in P}. \tag{9}$$

With our definition, the flow formulation in Eq. (5) becomes:

$$\begin{aligned} \phi_t^N(x_0|x_1) &= tNN(x_0, x_1) + (1-t)x_0, \\ v_t^N(x_0|x_1) &= NN(x_0, x_1) - x_0, \end{aligned} \tag{10}$$

and the loss in Eq. (7) becomes:

$$\mathcal{L}_{NFM} = \mathbb{E}_{t,x_0,x_1,\tilde{c}} \left[\left\| u_\theta(t, \phi_t^N(x_0|x_1), \tilde{c}) - v_t^N(x_0|x_1) \right\|^2 \right]. \tag{11}$$

4.3 Chamfer Distance Matching

NFM does not ensure one-to-one correspondences, as multiple points in x_0 may map to the same point in x_1. This can produce discrete reconstructions with reduced scene occupancy and lower fine-grained point density. To address this, we introduce the Chamfer Distance Matching (CDM) loss based on the target objective in Eq. (7):

$$\mathcal{L}_{CDM} = \mathbb{E}_{t,x_0,x_1,\tilde{c}} \left[CD(x_0 + u_\theta(t, \phi_t^N(x_0|x_1), \tilde{c}), x_1) \right], \tag{12}$$

where the Chamfer Distance (CD) is defined as:

$$CD(P, \hat{P}) = \frac{1}{|P|} \sum_{\mathbf{p} \in P} \min_{\hat{\mathbf{p}} \in \hat{P}} \|\mathbf{p} - \hat{\mathbf{p}}\|^2 + \frac{1}{|\hat{P}|} \sum_{\hat{\mathbf{p}} \in \hat{P}} \min_{\mathbf{p} \in P} \|\mathbf{p} - \hat{\mathbf{p}}\|^2, \tag{13}$$

and promotes a point-wise matching between x_0 and x_1. CDM encourages the model to predict vector fields that spread out the generated point cloud to cover the entire target scene by accounting for mutual alignment between x_0 and x_1.

4.4 Training and Inference

Training. During training, given a target data x_1 and the corresponding single LiDAR scan P, we compute x_0 with Eq. (8) and uniformly sample $t \in [0, 1]$. The condition $\tilde{c}$ is sampled according to the classifier-free guidance [6] scheme. Following Eq. (10), we compute the conditional flow and vector field, and compute the loss:

$$\mathcal{L} = \lambda_{\text{NFM}} \mathcal{L}_{NFM} + \lambda_{\text{CDM}} \mathcal{L}_{CDM}, \tag{14}$$

where $\mathcal{L}_{NFM}$ establishes the point-wise flow matching, and $\mathcal{L}_{CDM}$ encourages the generated point cloud to spread out and align with the target complete scene. λ_{NFM} and λ_{CDM} represent the respective loss weights.

Inference. During inference, given a single LiDAR scan P, we compute x_0 with Eq. (8), and we progressively follow the flow $t\colon 0 \to 1$ using the vector field predicted by the model using Euler's method [14]:

$$\begin{aligned} &X_0 = x_0, \\ &X_{t+h} = X_t + h u^*_\theta(t, X_t, P) \ (t = 0, h, 2h, \ldots, 1 - h), \end{aligned} \tag{15}$$

where h is the step size and $u^*_\theta(t, X_t, P)$ is obtained through classifier-free guidance [6]:

$$u^*_\theta(t, X_t, P) = u_\theta(t, X_t, \emptyset) + w[u_\theta(t, X_t, P) - u_\theta(t, X_t, \emptyset)], \tag{16}$$

with w the classifier-free conditioning weight.

5 Experiments

5.1 Experimental Settings

Datasets. We train LiFlow on the SemanticKITTI dataset [1] using sequences $00-07$ and $09-10$, and evaluate on sequence 08. We further evaluate generalization on sequence 00 of the Apollo Columbia Park MapData dataset [17]. This dataset is recorded with the same sensor, but in a different environment.

We follow Nunes *et al.* [24] to generate the ground truth complete scans, using the dataset poses to aggregate the scans in the sequence and remove moving objects to build a map for each sequence. Moving objects are removed using the ground truth semantic labels for SemanticKITTI and the labels provided by Chen *et al.*; Mersch *et al.* [2, 23] for the Apollo dataset. For both the datasets, we set the scan range to 50 m.

Training. We train LiFlow for 20 epochs, using only the training set from SemanticKITTI. We set the classifier-free probability [6] $p = 0.1$. The loss in Eq. (14) uses $\lambda_{\text{NFM}} = 1$ and $\lambda_{\text{CDM}} = 0.1$. The network u_θ adopts the MinkUNet [3] architecture used in LiDPM [21]. This architecture is the same as the one proposed in LiDiff [24], except that all Batch Normalization (BN) [7] layers are replaced by Instance Normalization (IN) [34] layers. We set the quantization resolution of the MinkUNet u_θ to 0.05 m and we train the network with Exponential Moving Average (EMA) [32] with decay $\alpha = 0.9999$. The training optimization is implemented with Adam [9] and the batch size is set to 4. For each input scan, we sample $N = 18,000$ points with farthest point sampling. For the ground truth, we randomly sample $M = 180,000$ points without replacement, *i.e.*, $K = 10$.

Training and inference are performed using an NVIDIA A100 GPU with 80 GB of memory.

Inference. For inference, we use Euler's method [14] for 10 inference steps, *i.e.*, $h = 0.1$. We set the classifier-free conditioning weight [6] to $w = 6.0$. We further validate LiFlow using the refinement network from Nunes *et al.* [24] to refine the generated complete scenes X_1 and scale up the number of points by a factor $\kappa = 6$, *i.e.*, $X_1^\dagger \in \mathbb{R}^{O\times 3}, O = \kappa M$. We use the official code of the refinement network and the provided weights also trained on SemanticKITTI.

Metrics. We evaluate and compare LiFlow using the Chamfer Distance (CD), the Jensen-Shannon Divergence (JSD) and the Voxel Intersection over Union (Voxel IoU). The CD and JSD are two common metrics to evaluate the reconstruction fidelity of 3D point clouds [24,26,38]. The CD evaluates the completion at the point level, measuring the level of detail of the reconstructed scene by calculating how far its points are from the ground truth scene. The JSD is a statistical metric that compares the point distribution between the reconstruction and the ground truth scene. For the JSD, we follow the evaluation proposed

by Xiong *et al*. [38], where the scene is first voxelized with a grid resolution of 0.5 m and projected to a Bird's Eye View (BEV). The Voxel IoU evaluates the occupancy of the reconstructed scene compared with the ground truth scene after voxelizing them. We follow the evaluation proposed by Song *et al*. [30] and classify the voxel occupancy at three different voxel resolutions, *i.e*., 0.5 m, 0.2 m, and 0.1 m. While using a voxel resolution of 0.5 m, the Voxel IoU evaluates the occupancy over the coarse scene, for decreasing voxel resolutions, more fine-grained details are considered.

Baselines. For LMSCNet [27], LODE [12], MID [35], and PVD [40], we report the results from Nunes *et al*. [24] on the SemanticKITTI dataset. For LiDiff [24] and LiDPM [21], we use their official code and the provided weights also trained on SemanticKITTI.

5.2 Scene Completion

Table 1 compares LiFlow with state-of-the-art methods on the SemanticKITTI validation set. Point-level methods (LiDiff, LiDPM, LiFlow) outperform voxel-level methods (LMSCNet, LODE, MID, PVD) in CD. With the refinement network, LiFlow achieves the best CD results and leads in JSD, demonstrating superior reconstruction fidelity. Our method also delivers competitive performance in scene occupancy, as measured by Voxel IoU. While LiDiff and LiDPM achieve these results in 50 and 20 inference steps of DPM-Solver [16], respectively, LiFlow only requires 10 inference steps of Euler's method [14].

Figure 2 shows a qualitative comparison of LiFlow with LiDiff and LiDPM on a single LiDAR scan from the SemanticKITTI validation set (50 inference steps). With the refinement network, LiFlow exhibits a more stable reconstruction and

Table 1: Evaluation on the SemanticKITTI validation set (sequence 08). Best and second-best results are highlighted in bold and underlined, respectively. †: with refinement network [24].

Method	CD[m]↓	JSD[m]↓	Voxel IoU[%]↑		
			0.5	0.2	0.1
LMSCNet [27]	0.641	0.431	30.8	12.1	3.7
LODE [12]	1.029	0.451	33.8	16.4	5.0
MID [35]	0.503	0.470	31.6	22.7	13.1
PVD [40]	1.256	0.498	15.9	4.0	0.6
LiDiff [24]	0.434	0.444	31.5	16.8	4.7
LiDiff† [24]	0.375	0.416	32.4	23.0	<u>13.4</u>
LiDPM [21]	0.446	0.440	34.1	19.5	6.3
LiDPM† [21]	0.377	<u>0.403</u>	<u>36.6</u>	**25.8**	**14.9**
LiFlow (ours)	<u>0.309</u>	0.416	31.6	13.1	3.8
LiFlow† (ours)	**0.228**	**0.367**	**37.2**	<u>25.1</u>	12.9

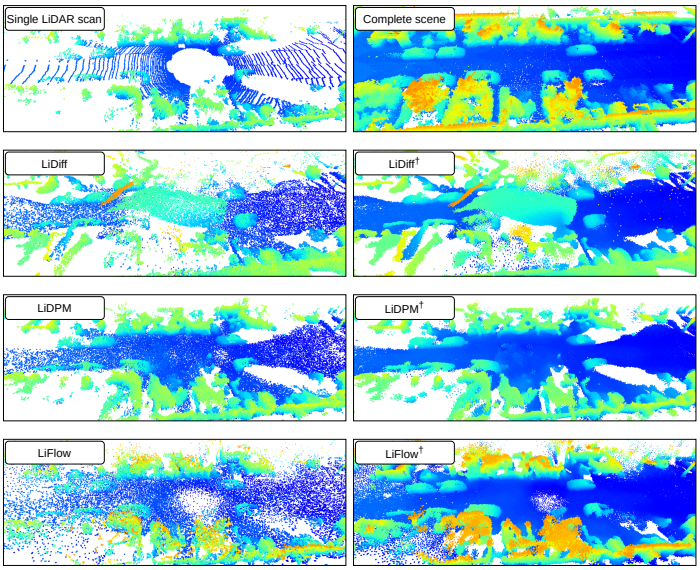

Fig. 2: Qualitative results on a single LiDAR scan from the SemanticKITTI validation set (sequence 08). Colors depict point height. †: with refinement network [24].

enhanced geometric completion of the scene compared to the other methods, thanks to the consistent distribution between training and inference.

Table 2 compares LiFlow with LiDiff and LiDPM on the Apollo dataset at multiple inference steps (5, 10, 20, and 50) using Euler's method and DPM-Solver. LiFlow demonstrates superior generalization to unseen environments and stable performance across all inference step counts. This advantage arises from our flow matching formulation, which aligns the target distribution with the OT path to accelerate convergence. By reducing the number of inference steps, LiFlow enables faster inference compared to existing diffusion-based methods, while preserving stability and high generation quality.

Figure 3 shows a qualitative comparison of LiFlow with LiDiff and LiDPM on a single LiDAR scan from the Apollo dataset (50 inference steps). With the refinement network, LiFlow achieves superior geometric completion, *e.g.*, of cars, demonstrating state-of-the-art scene completion performance.

5.3 Ablation Studies

Table 3 presents an ablation study on the loss weights λ_{NFM} and λ_{CDM} on the Apollo dataset. Using only NFM increases the CD due to the lack of direct CD supervision, coarser voxel IoU (0.5, 0.2) improves, but fine-grained occupancy (0.1) drops as point dispersion is not encouraged. Using only CDM reduces overall performance, particularly Voxel IoU, since NFM constitutes the foundation

Table 2: Evaluation on the Apollo dataset (sequence 00). Each method is evaluated at multiple inference steps. Inference time per frame is also reported. Best and second-best results for each method are highlighted in bold and underlined, respectively. †: with refinement network [24].

Method	Steps	CD[m]↓	JSD[m]↓	Voxel IoU[%]↑			Time[s]
				0.5	0.2	0.1	
LiDiff [24]	5	0.492	**0.381**	21.3	5.4	1.1	5.1
	10	**0.437**	0.398	**26.7**	9.2	2.3	7.4
	20	0.457	0.427	26.3	11.3	3.3	12.3
	50	0.451	0.430	25.6	**13.4**	**4.4**	27.1
LiDiff† [24]	5	0.396	**0.322**	27.6	12.1	3.7	5.2
	10	**0.350**	0.343	**33.4**	17.8	6.7	7.5
	20	0.386	0.380	31.3	19.0	9.2	12.4
	50	0.389	0.381	29.5	**20.2**	**11.8**	27.2
LiDPM [21]	5	0.482	0.440	27.8	14.5	**4.7**	5.1
	10	0.477	0.438	28.4	14.9	4.6	7.4
	20	**0.476**	0.437	29.0	15.2	4.6	12.3
	50	**0.476**	**0.436**	**29.4**	**15.3**	4.5	27.1
LiDPM† [21]	5	0.413	0.403	30.5	21.2	12.4	5.2
	10	**0.406**	**0.400**	31.1	21.6	12.6	7.5
	20	**0.406**	**0.400**	**31.3**	21.8	**12.7**	12.4
	50	0.408	0.401	**31.3**	**21.9**	**12.7**	27.2
LiFlow (ours)	5	**0.351**	**0.398**	**28.2**	**10.7**	**3.0**	4.7
	10	0.359	0.406	27.2	10.4	2.8	7.3
	20	0.365	0.409	26.8	10.0	2.7	12.7
	50	0.367	0.412	26.6	9.8	2.6	28.8
LiFlow† (ours)	5	**0.277**	**0.346**	**33.4**	**21.6**	11.0	4.8
	10	0.282	0.350	32.9	21.4	**11.1**	7.4
	20	0.285	0.351	32.6	21.1	10.8	12.8
	50	0.286	0.352	32.4	20.8	10.6	28.9

of the iterative generative process. Equal weighting of both losses yields performance comparable to the baseline, with CDM encouraging greater point spread, highlighting the complementary roles of NFM and CDM in the proposed FM formulation.

Table 4 presents an ablation study on the MinkUNet u_θ of LiFlow with BN [7] layers and IN [34] layers on the Apollo dataset. We observe that the use of IN layers significantly improves overall performance across all metrics and leads to substantial improvements in the occupancy of the reconstructed scenes.

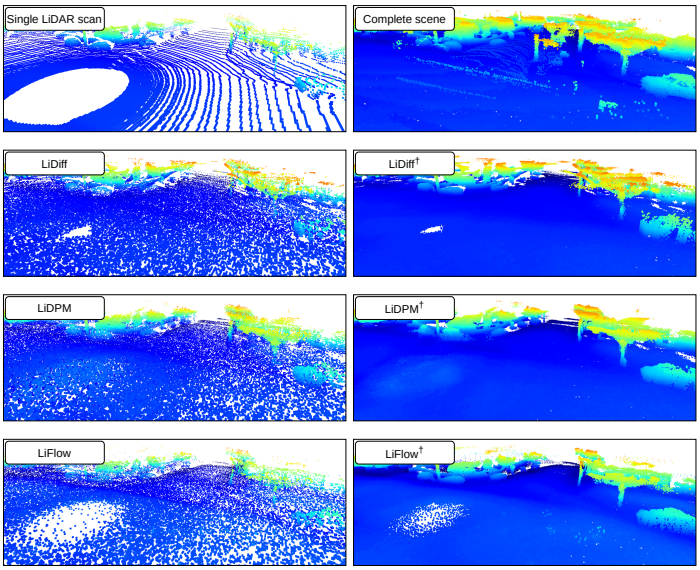

Fig. 3: Qualitative results on a single LiDAR scan from the Apollo dataset (sequence 00). Colors depict point height. †: with refinement network [24].

Table 3: Ablation study on LiFlow for different loss weights (λ_{NFM}, λ_{CDM}) on the Apollo dataset (sequence 00). The first block corresponds to training with only the NFM loss ($\lambda_{\text{CDM}} = 0$), the second with only the CDM loss ($\lambda_{\text{NFM}} = 0$), the third with equal weighting, and the last block reports the baseline configuration. Best and second-best results are highlighted in bold and underlined, respectively. †: with refinement network [24].

			CD[m]↓	JSD[m]↓	Voxel IoU[%]↑		
λ_{NFM}	λ_{CDM}				0.5	0.2	0.1
1	0		0.411	0.414	29.0	10.5	2.8
1	0	†	0.325	<u>0.356</u>	**34.8**	**22.3**	<u>10.6</u>
0	1		0.386	0.417	21.6	7.7	2.4
0	1	†	0.323	0.397	25.3	13.8	6.7
1	1		0.355	0.397	26.5	9.7	2.7
1	1	†	<u>0.285</u>	0.364	31.4	19.7	10.0
1	0.1		0.359	0.406	27.2	10.4	2.8
1	0.1	†	**0.282**	**0.350**	<u>32.9</u>	<u>21.4</u>	**11.1**

Table 4: Ablation study on the MinkUNet u_θ of LiFlow with BN layers and IN layers on the Apollo dataset (sequence 00). Best results for each block are highlighted in bold. †: with refinement network [24].

Method	CD[m]↓	JSD[m]↓	Voxel IoU[%]↑		
			0.5	0.2	0.1
BN	0.405	**0.397**	24.9	7.7	1.8
IN	**0.359**	0.406	**27.2**	**10.4**	**2.8**
BN†	0.340	0.354	29.0	16.3	7.0
IN†	**0.282**	**0.350**	**32.9**	**21.4**	**11.1**

6 Conclusion

Summary. We proposed LiFlow as the first flow matching method for 3D LiDAR scene completion at the point level. By directly aligning single LiDAR scans to complete scenes, LiFlow overcomes the distribution mismatch in existing diffusion-based methods. Extensive experiments demonstrate its effectiveness, achieving state-of-the-art performance across multiple metrics while requiring fewer inference steps than existing diffusion-based methods.

Limitations. While LiFlow achieves consistent improvements in CD, JSD, and coarse occupancy, fine-grained Voxel IoU remains challenging. This limitation appears inherent to the lack of one-to-one matching between the initial and target point clouds. Although the proposed CDM loss encourages better point-wise correspondences, the NFM formulation still lacks explicit constraints for establishing one-to-one matches. Addressing this limitation and improving fine-grained occupancy remains an important direction for future work.

References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9297–9307 (2019)
2. Chen, X., Mersch, B., Nunes, L., Marcuzzi, R., Vizzo, I., Behley, J., Stachniss, C.: Automatic labeling to generate training data for online lidar-based moving object segmentation. IEEE Robotics and Automation Letters **7**(3), 6107–6114 (2022)
3. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)
4. Fu, C., Dong, C., Mertz, C., Dolan, J.M.: Depth completion via inductive fusion of planar lidar and monocular camera. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10843–10848. IEEE (2020)

5. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
6. Ho, J., Salimans, T.: Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications (2021)
7. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. pmlr (2015)
8. Jaritz, M., De Charette, R., Wirbel, E., Perrotton, X., Nashashibi, F.: Sparse and dense data with cnns: Depth completion and semantic segmentation. In: 2018 International Conference on 3D Vision (3DV). pp. 52–60. IEEE (2018)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
10. Lan, Y., Zhou, S., Lyu, Z., Hong, F., Yang, S., Dai, B., Pan, X., Loy, C.C.: Gaussiananything: Interactive point cloud flow matching for 3d generation. In: International Conference on Learning Representations (2025)
11. Lee, J., Im, W., Lee, S., Yoon, S.E.: Diffusion probabilistic models for scene-scale 3d categorical data. arXiv preprint arXiv:2301.00527 (2023)
12. Li, P., Zhao, R., Shi, Y., Zhao, H., Yuan, J., Zhou, G., Zhang, Y.Q.: Lode: Locally conditioned eikonal implicit scene completion from sparse lidar. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 8269–8276. IEEE (2023)
13. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: International Conference on Learning Representations (2023)
14. Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R.T., Lopez-Paz, D., Ben-Hamu, H., Gat, I.: Flow matching guide and code. arXiv preprint arXiv:2412.06264 (2024)
15. Liu, Z.S., He, C., Ju, Y., Li, L.: Pufm: Efficient point cloud upsampling via flow matching. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 7458–7466 (2026)
16. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. Advances in neural information processing systems **35**, 5775–5787 (2022)
17. Lu, W., Zhou, Y., Wan, G., Hou, S., Song, S.: L3-net: Towards learning based lidar localization for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6389–6398 (2019)
18. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2837–2845 (2021)
19. Ma, F., Cavalheiro, G.V., Karaman, S.: Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: 2019 international conference on robotics and automation (ICRA). pp. 3288–3295. IEEE (2019)
20. Ma, F., Karaman, S.: Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: 2018 IEEE international conference on robotics and automation (ICRA). pp. 4796–4803. IEEE (2018)
21. Martyniuk, T., Puy, G., Boulch, A., Marlet, R., de Charette, R.: Lidpm: Rethinking point diffusion for lidar scene completion. In: 2025 IEEE Intelligent Vehicles Symposium (IV). pp. 555–560. IEEE (2025)

22. Matteazzi, A., Colling, P., Arnold, M., Tutsch, D.: A preprocessing and postprocessing method for lidar semantic segmentation in long distance. In: International Conference on Neural Information Processing. pp. 288–304. Springer (2025)
23. Mersch, B., Guadagnino, T., Chen, X., Vizzo, I., Behley, J., Stachniss, C.: Building volumetric beliefs for dynamic environments exploiting map-based moving object segmentation. IEEE Robotics and Automation Letters **8**(8), 5180–5187 (2023)
24. Nunes, L., Marcuzzi, R., Mersch, B., Behley, J., Stachniss, C.: Scaling diffusion models to real-world 3d lidar scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14770–14780 (2024)
25. Popović, M., Thomas, F., Papatheodorou, S., Funk, N., Vidal-Calleja, T., Leutenegger, S.: Volumetric occupancy mapping with probabilistic depth completion for robotic navigation. IEEE Robotics and Automation Letters **6**, 5072–5079 (2021)
26. Ran, H., Guizilini, V., Wang, Y.: Towards realistic scene generation with lidar diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14738–14748 (2024)
27. Roldao, L., De Charette, R., Verroust-Blondet, A.: Lmscnet: Lightweight multiscale 3d semantic completion. In: 2020 International Conference on 3D Vision (3DV). pp. 111–119. IEEE (2020)
28. Schusterbauer, J., Gui, M., Fundel, F., Ommer, B.: Diff2flow: Training flow matching models via diffusion model alignment. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 28347–28357 (2025)
29. Schusterbauer, J., Gui, M., Ma, P., Stracke, N., Baumann, S.A., Hu, V.T., Ommer, B.: Fmboost: Boosting latent diffusion with flow matching. In: European Conference on Computer Vision. pp. 338–355. Springer (2024)
30. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1746–1754 (2017)
31. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021)
32. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems **30** (2017)
33. Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with minibatch optimal transport. Transactions on Machine Learning Research pp. 1–34 (2024)
34. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv preprint arXiv:1607.08022 (2016)
35. Vizzo, I., Mersch, B., Marcuzzi, R., Wiesmann, L., Behley, J., Stachniss, C.: Make it dense: Self-supervised geometric scan completion of sparse 3d lidar scans in large outdoor environments. IEEE Robotics and Automation Letters **7**(3), 8534–8541 (2022)
36. Wu, X., Peng, L., Yang, H., Xie, L., Huang, C., Deng, C., Liu, H., Cai, D.: Sparse fuse dense: Towards high quality 3d detection with depth completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5418–5427 (2022)
37. Xia, Z., Liu, Y., Li, X., Zhu, X., Ma, Y., Li, Y., Hou, Y., Qiao, Y.: Scpnet: Semantic scene completion on point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17642–17651 (2023)

38. Xiong, Y., Ma, W.C., Wang, J., Urtasun, R.: Learning compact representations for lidar completion and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1074–1083 (2023)
39. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 3101–3109 (2021)
40. Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5826–5835 (2021)