SUMI: SYMMETRY-GUIDED UPSAMPLING MODULE WITH ITERATIVE DIFFUSION FOR POINT CLOUD COMPLETION

Anonymous authors
Paper under double-blind review

ABSTRACT

Point cloud completion commonly follows a coarse-to-fine paradigm, where a low-density coarse shape is first predicted and then upsampled to the target resolution. Although recent methods have improved global structure recovery, the fine stage often remains limited by simple upsampling and insufficient interaction with coarse structural features, making local detail reconstruction challenging. We propose SUMI, a diffusion-enhanced refinement module for coarse-to-fine point cloud completion. Unlike prior diffusion-based completion methods that use diffusion as a standalone point generator, SUMI injects noisy geometric features into cross-attention with coarse structural features, enabling reverse denoising to refine local geometry while preserving global consistency. SUMI can also be integrated into existing coarse-to-fine models as a flexible refinement module. Experiments on PCN, ShapeNet-55/34, and MVP demonstrate consistent improvements over strong baselines. SUMI achieves the best overall CD and F1-score on PCN, reduces CD by up to 16.1% on ShapeNet-55, and obtains the best CD across all output densities on MVP. The anonymous repo can be accessed here: https://anonymous.4open.science/r/Sumi-2278/

1 INTRODUCTION

Point clouds provide a flexible and efficient representation of 3D geometry captured by LiDAR, RGB-D cameras, and multi-view systems, supporting autonomous driving, robotics, AR/VR, digital twins, and 3D content creation. Limited fields of view, occlusion, sparse sampling, and noise often produce incomplete observations that impair recognition, segmentation, and reconstruction. Point cloud completion aims to recover complete, dense, and geometrically consistent shapes from such partial inputs.

Early point cloud completion methods mainly relied on traditional techniques such as geometric priors, rule constraints, or template matching Kazhdan et al. (2006); Li et al. (2015); Pauly et al. (2005). These methods typically assume that objects have specific structural patterns or topological priors, and can achieve certain results in simple scenes. However, their generalization ability and robustness are significantly limited when faced with complex object shapes or cross-class scenes. With the development of deep learning, researchers have begun to utilize data-driven methods to learn implicit or explicit shape priors, significantly advancing the field of point cloud completion Han et al. (2017); Yu et al. (2021); Wang et al. (2021; 2018). These methods typically map a portion of the point cloud to a latent representation space using an encoder, and then generate the complete point cloud using a decoder, achieving significant improvements in completion accuracy and stability. Recent advances in representation learning have introduced more expressive architectures, such as attention mechanisms, masked modeling, and generative frameworks, to better capture long-range dependencies and global shape priors. By learning from partially observed data and exploiting large-scale pretraining or generative sampling, these approaches show improved robustness and diversity in completion results.

Many 3D objects exhibit geometric symmetry, making symmetry priors useful for point cloud completion. While early symmetry-based methods relied mainly on global transformations, they often struggled to recover accurate local geometric relationships. Recent work Yan et al. (2025) addresses

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

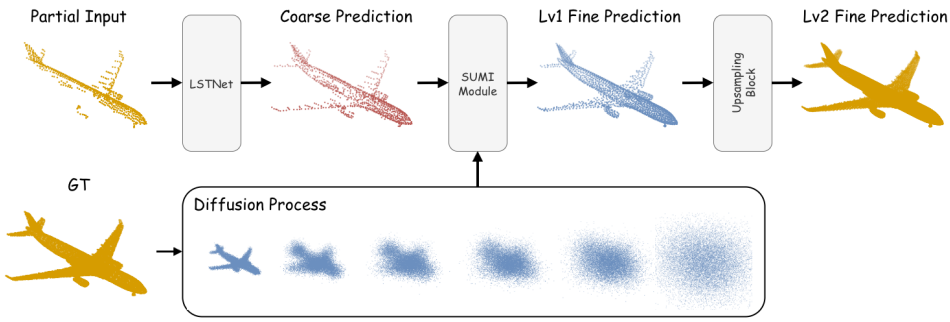


Figure 1: Overview of SUMI. A coarse prediction is first generated from the partial input, followed by diffusion-enhanced Level-1 refinement and lightweight upsampling for the final output.

this by introducing LSTNet, which leverages local symmetry information to improve missing-region reconstruction. Nevertheless, most deep learning-based methods still adopt a coarse-to-fine pipeline, where a low-density coarse output is further upsampled to the target resolution. This makes final completion quality dependent on the upsampling stage and the limited information carried by the coarse representation.

To address the aforementioned issues, we propose SUMI, a diffusion-enhanced refinement mechanism within the coarse-to-fine framework for high-quality point cloud completion. Different from prior diffusion-based completion methods that directly generate point coordinates through a standalone denoising process, SUMI uses diffusion noise as a fine-stage refinement signal inside feature interaction. By injecting noisy geometric features into cross-attention with the coarse representation, SUMI progressively enhances local geometric details while preserving the global structure provided by the coarse prediction. Therefore, SUMI is not designed to replace the coarse-to-fine pipeline, but to strengthen its fine-generation stage and can be integrated into existing coarse-to-fine completion models as a flexible refinement module. Our main contributions are:

- We propose SUMI, a symmetry-guided conditional diffusion module for refinement between coarse prediction and dense upsampling. Coarse structural queries attend to evolving noisy geometry, then fuse with partial and symmetry features to predict refined points.
- On PCN, diffusion refinement outperforms trained iterative deterministic SUMI at the same 200-step count, supporting benefits beyond iteration count in this setting. Further ablations examine geometric conditioning and refinement-stage placement.
- Integrating SUMI improves completion accuracy in AdaPoinTr and the SymmCompletion-based pipeline. Evaluations on PCN, ShapeNet-55/34, and MVP demonstrate effectiveness across object categories, missingness levels, and output densities.

2 Related Work

2.1 Point Cloud Completion (PCC)

Early PCC approaches primarily relied on geometric priors, template matching, or symmetry assumptions to recover missing regions Kazhdan et al. (2006); Yang et al. (2017); Li et al. (2015); Pauly et al. (2005); Kazhdan et al. (2004); Mitra et al. (2006), which limited their applicability to simple shapes and constrained scenarios. With the advancement of deep learning, PCC methods have evolved into two different types. The first type focuses on predicting globally completed point clouds directly from partial input, typically by encoding the incomplete observation into a latent representation that captures overall shape structure, followed by decoding to generate a complete point set Yang et al. (2018); Fu et al. (2023); Wang et al. (2024). These methods benefit from strong global shape reasoning but may suffer from loss of fine-grained geometric details. The second type directly focuses on inferring missing regions conditioned on visible points, explicitly modeling the relationship between observed and unobserved geometry Yu et al. (2021); Zhou et al. (2022).

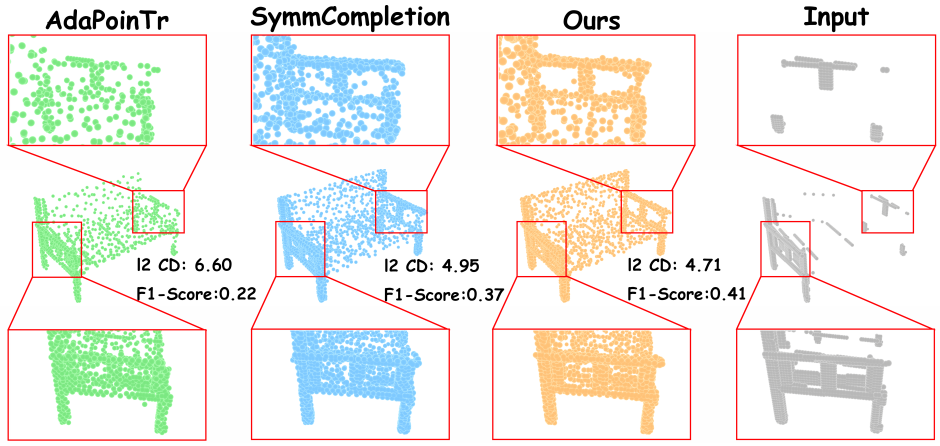


Figure 2: Level-1 fine prediction comparison. From left to right: AdaPoinTr, SymmCompletion, Ours, and the partial input. SUMI reconstructs better local details and more uniform point distributions. Per-sample L_2 Chamfer Distance $\times 10^4$ (CD, $\downarrow$) and F1-Score@1% (F1, $\uparrow$) are shown.

2.2 TRANSFORMER

Transformers, originally introduced for NLP Vaswani et al. (2017), leverage self-attention to model long-range dependencies and global context, achieving strong performance in language understanding and generation Devlin et al. (2019); OpenAI (2024). Their success has driven adoption in computer vision, where ViT Dosovitskiy (2020) represents images as token sequences and has shown competitive results across classification, detection, and segmentation tasks Touvron et al. (2021); He et al. (2022); Manzari et al. (2023); Li et al. (2023); Zong et al. (2023); Zhao et al. (2024); Lv et al. (2024b); Liu et al. (2025a); Kumar et al. (2024); Hu et al. (2024); Liu et al. (2025b). Recently, Transformers have been extended to 3D tasks, including point cloud classification, segmentation, and generation, demonstrating strong global reasoning and robustness Long et al. (2023); Wu et al. (2025a); Liu et al. (2025c); Kolodiazhnyi et al. (2024); Xiao et al. (2025); Li et al. (2025a); Lu et al. (2025); Wu et al. (2024); Lv et al. (2024a). In point cloud completion, PoinTr Yu et al. (2021) and subsequent works Zhou et al. (2022); Wen et al. (2022); Li et al. (2025b) establish the effectiveness of Transformer-based architectures.

2.3 DIFFUSION MODEL (DM)

DMs have emerged as powerful generative frameworks that iteratively add and remove noise, enabling stable training and high-quality synthesis Sohl-Dickstein et al. (2015); Ho et al. (2020). Initially developed for image generation Croitoru et al. (2023), they have achieved strong performance across tasks such as image generation, super-resolution, and inpainting Batzolis et al. (2021); Yu et al. (2023); Zhu et al. (2023a); Li et al. (2024); Gao et al. (2023); Shang et al. (2024); Xiao et al. (2023); Niu et al. (2024); Wang et al. (2025b); Kim et al. (2025). Recent work extends DMs to 3D tasks, including point cloud, shape, and scene generation Romanelis et al. (2025); Luo & Hu (2021); Wu et al. (2023); Vahdat et al. (2022); Xiong et al. (2025); Zhang et al. (2023); Zhou et al. (2021); Liu et al. (2024); Wang et al. (2025a); Ju et al. (2024). By modeling complex geometric distributions, DMs are well suited for irregular point clouds and enable progressive refinement of structure and detail. In PCC, diffusion-based methods were first introduced by PVD Zhou et al. (2021), followed by subsequent extensions Kasten et al. (2023); Du et al. (2025). PDR Lyu et al. (2022) further explores conditional diffusion refinement for point cloud completion. In contrast, SUMI embeds noisy geometric information into feature-level cross-attention and uses diffusion as a fine-stage refinement signal within a coarse-to-fine framework.

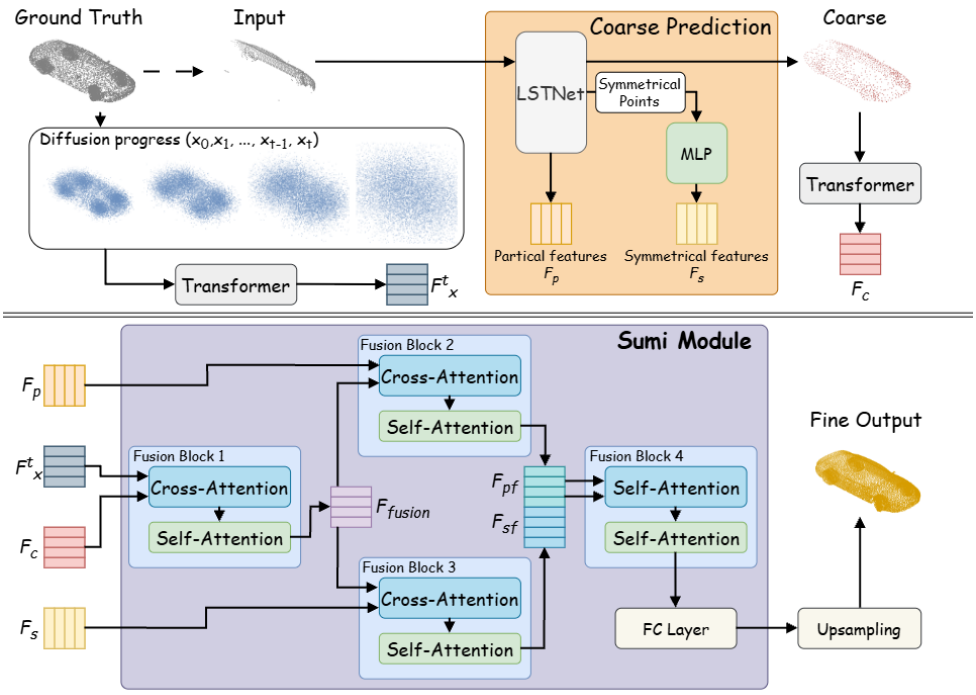


Figure 3: Detailed architecture of the SUMI module. The coarse prediction and noisy point cloud are encoded and fused through noise-conditioned cross-attention, followed by partial and symmetry feature fusion for fine prediction.

3 METHODOLOGY

3.1 OVERVIEW

Fig. 1 presents the overview of our three-layer coarse-to-fine pipeline, which consists of one coarse prediction layer and two fine-generation layers. The coarse layer first predicts a low-density complete shape from the partial input and extracts structural features from the input and its symmetric prediction. During training, Gaussian noise is progressively added to a uniformly downsampled ground-truth point cloud, and the noisy point cloud is fused with the coarse representation through Transformer-based feature interaction. This enables SUMI to learn a noise-conditioned refinement process that recovers fine geometric details while preserving global structure. During inference, SUMI performs T reverse-denoising steps to produce the Level-1 fine prediction, which is passed to SGFormer for final dense upsampling.

3.2 COARSE GENERATION

For the coarse generation, we take the standard LSTNet module Yan et al. (2025) to make the coarse prediction. This module is based on a local point-wise symmetry transformation that maps the existing geometric structure in the partial point cloud to the missing region, thereby constructing an initial point cloud representation with high consistency and fidelity. LSTNet outputs coarse keypoints P_c, partial features F_p, and symmetry features F_s. Keypoint guidance uses P_c through its encoded features F_c and is distinct from partial-feature guidance.

3.3 SUMI MODULE

We apply the SUMI Module for Level-1 Fine generation. We demonstrate the performance of our proposed method for the Level-1 Fine generation and comparison with prior methods in Fig. 2.

3.3.1 DIFFUSION PROCESS

SUMI is based on the standard diffusion framework. We apply diffusion noise in the point-coordinate space before feature encoding, allowing the model to learn recovery from noisy geometry through attention-based feature interaction. Let P_{gt} denote the complete ground-truth point cloud, and let x_0 be the 1,024-point clean target obtained by uniform downsampling of P_{gt}. During forward diffusion, Gaussian noise is added to x_0, producing states $x_1, \ldots, x_T$ with the same number of points. Following DDPM and prior works, we model this process as a Markov process as in Eq. 1:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

During training, we first fuse the coarse features and noise to enhance the model's ability to model complex distributions. Then, partial features F_p and symmetry features F_s are used as conditional information, along with the fused features, to predict the clean fine-level point cloud from the noisy observation. Through iterative optimisation, the model gradually recovers the target point cloud distribution from the noise.

For the main configuration, the denoising branch predicts a 1,024-point clean estimate $\hat{x}_{0,t}$ from the 1,024-point state x_t, conditioned on the coarse proposal and geometric features. The Level-1 output x_f contains 2,048 points after merging the denoised points with the 1,024-point LSTNet proposal. Thus, $\hat{x}_{0,t}$ denotes the clean estimate for the reverse state, while x_f denotes the merged Level-1 output. The latter is supervised using Chamfer Distance:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{CD}(x_f, P_{\text{gt}}) \tag{2}$$

This objective is integrated into the end-to-end training objective in Section 3.5; Appendix I provides the diffusion formulation and training configuration.

3.3.2 FUSION BLOCK

In the SUMI Module, we include four Fusion Blocks for aggregating and enhancing features across multiple sources. We follow previous work Hong et al. (2023); Yan et al. (2025). The first 3 Fusion Blocks contain a Cross-Attention mechanism to aggregate features and a Self-Attention mechanism to enhance feature representation, and the last Fusion Block contains two Self-Attention mechanisms.

The overall pipeline of feature fusion inside the SUMI Module is shown in Fig. 3. The module takes as input the coarse prediction P_c, partial features F_p, symmetry features F_s, and the noisy point cloud at time step t, x_t.

We first encode P_c and x_t using two Transformer encoders to obtain feature representations F_c and F_x^t, respectively. These features are then fused through a cross-attention mechanism. Specifically, F_c is used as the Query, while F_x^t is used as the Key and Value, allowing the model to refine coarse structural features using noise-aware geometric information. That process is defined as below Eq. 3 and Eq. 4:

$$F_{fusion} = \mathcal{S}(\mathcal{C}(F_c, F_x^t)) \tag{3}$$

$$\mathcal{C}(F_c, F_x^t) = CrossAttention(Q = F_c, K = F_x^t, V = F_x^t) \tag{4}$$

where $\mathcal{S}$ refers to Self-Attention. Since the key and value are derived from noisy inputs x_t, the attention weights become implicitly noise-conditioned, allowing stochastic perturbations to influence feature interaction.

Next, we use two separate Fusion Blocks to incorporate partial and symmetry features. Specifically, F_{fusion} is fused with F_p to obtain F_{pf}, while another branch fuses F_{fusion} with F_s to obtain F_{sf}. The two resulting features are then concatenated and fed into the final Fusion Block, which uses self-attention to further enhance the overall feature representation. We define this process as Eq. 5:

$$F = \mathcal{S}(\mathcal{C}(F_{fusion}, F_p)) \cdot \mathcal{S}(\mathcal{C}(F_{fusion}, F_s)) \tag{5}$$

where $\cdot$ denotes the concatenation operation.

A fully connected layer maps the fused features to point coordinates. In the main configuration, SUMI denoises a 1,024-point state conditioned on the 1,024-point LSTNet proposal. After denoising, the resulting point cloud P_d is concatenated with the proposal along the point dimension to form the 2,048-point Level-1 output x_f (also denoted P_f):

$$x_f = \mathrm{Concat}_{\mathrm{points}}(P_c, P_d), \qquad |P_c| = |P_d| = 1{,}024. \tag{6}$$

The merged output is passed to SGFormer for final dense upsampling; it is not fed back into the reverse diffusion chain.

During inference, SUMI starts from a noisy point cloud and progressively refines it through the reverse denoising process. The clean-point parameterization of the reverse mean is summarized in Appendix I.3, and SUMI's reconstruction objective is given in Appendix I.4. At each timestep, the 1,024-point clean estimate $\hat{x}_{0,t}$ guides the transition from x_t to x_{t-1}, conditioned on the coarse output P_c, partial features F_p, and symmetry features F_s. The reverse state remains at 1,024 points throughout the T denoising steps; the 2,048-point Level-1 output is obtained by the subsequent merging operation.

Further implementation details and stage-placement considerations are provided in Appendix F.

3.4 Enhancement for existing methods

Based on the design of noise embedding in our SUMI Module, the forward and reverse diffusion processes can be integrated into coarse-to-fine completion methods with minimal architectural modifications to enhance feature representation and completion performance. We integrate SUMI into AdaPoinTr and evaluate the performance in Sec. 4.2. We further validate the performance of injecting the diffusion process into different modules of the existing coarse-to-fine structured model in Appendix C. These results suggest that SUMI can serve as a flexible refinement module for coarse-to-fine completion architectures.

3.5 Learning Target and Loss Function

Our proposed method employs a multi-module end-to-end training strategy, using the ground truth of the complete point cloud as the unified learning objective. During training, the results generated at each stage are co-optimised to improve the overall completion quality.

In the loss function design, we use Chamfer Distance as the primary metric to evaluate geometric consistency between the predicted and ground-truth point clouds. Considering the differences in density between outputs at different stages, we further construct a hierarchical composite loss function to jointly constrain the coarse output and results of different precision levels in the fine stage, thereby achieving collaborative optimisation across the entire process. The overall loss function form is shown below:

$$\mathcal{L} = \mathcal{L}_{CD}(P_c, P_{\mathrm{gt}}) + \mathcal{L}_{CD}(x_f, P_{\mathrm{gt}}) + \mathcal{L}_{CD}(P_f'', P_{\mathrm{gt}}) \tag{7}$$

Here P_{gt} is the complete ground truth, P_c is the coarse prediction, x_f is the merged Level-1 prediction, and P_f'' is the final dense output. The 1,024-point x_0 is used for forward noising and is distinguished from the reconstruction target P_{gt}. This multi-scale supervision mechanism can simultaneously constrain the global structure and local details, thereby effectively improving the model's performance.

4 Experiments

In this section, we first introduce the dataset and benchmark for the PCC task. Then, we present the results of our model and compare them with several baselines. We also include ablation studies and visual analysis for our model. We train and validate our method on a single NVIDIA RTX4090.

Table 1: Quantitative results in terms of $l1$ Chamfer Distance $\times 10^3$(CD) and F1-Score@1% (F1) on PCN dataset.

Methods	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	CD (↓)	F1 (↑)
PCNYuan et al. (2018)	5.50	22.7	10.63	8.70	11.00	11.34	11.68	8.59	9.64	0.695
AdaPoinTrYu et al. (2021)	3.68	8.82	7.47	6.85	5.47	8.35	5.80	5.76	6.53	-
SVDFormerZhu et al. (2023b)	3.62	8.79	7.46	6.91	5.33	8.49	5.90	5.83	6.54	0.841
CRA-PCNRong et al. (2024)	3.59	8.70	7.50	6.70	5.06	8.24	5.72	5.64	6.39	-
T-CorresNetDuan et al. (2024)	3.63	9.79	7.47	6.85	5.47	8.35	5.80	5.76	6.53	0.845
DC-PCNWu et al. (2025b)	3.65	8.75	7.48	6.71	5.35	8.28	5.76	5.71	6.46	0.850
SymmCompletionYan et al. (2025)	3.56	8.51	7.34	6.56	5.09	8.40	5.71	5.52	6.33	0.852
AdaPoinTr + Ours	3.56	8.80	7.43	6.87	5.24	8.42	5.91	5.67	6.48	-
Ours	**3.53**	**8.50**	**7.30**	**6.55**	**4.95**	8.25	**5.65**	**5.52**	**6.27**	**0.855**

Table 2: Quantitative results in terms of $l2$ Chamfer Distance $\times 10^3$(CD, ↓) on ShapeNet55/34 dataset for three difficulty levels.

Method	ShapeNet-55			ShapeNet-34			Unseen 21		
	CD-S	CD-M	CD-H	CD-S	CD-M	CD-H	CD-S	CD-M	CD-H
PCNYuan et al. (2018)	1.94	1.96	4.08	1.87	1.81	2.97	3.17	3.08	5.29
PoinTrYu et al. (2021)	0.58	0.88	1.79	0.76	1.05	1.88	1.04	1.67	3.44
AdaPoinTrYu et al. (2021)	0.49	0.69	1.24	0.48	0.63	1.07	0.61	0.96	2.11
SeedFormerZhou et al. (2022)	0.50	0.88	1.79	0.48	0.70	1.30	0.61	1.07	2.35
SVDFormerZhu et al. (2023b)	0.48	0.70	1.30	0.46	0.64	1.13	0.61	1.05	2.19
CRA-PCNRong et al. (2024)	0.48	0.71	1.37	0.45	0.65	1.18	0.55	0.97	2.19
SymmCompletionYan et al. (2025)	0.36	0.55	1.12	0.33	0.48	1.00	**0.39**	0.70	1.83
Ours	**0.32**	**0.47**	**0.94**	**0.33**	**0.47**	**0.91**	0.43	**0.70**	**1.70**

4.1 Experimental Setup and Evaluation Metrics

We use PCN Yuan et al. (2018) as the primary dataset for training and evaluation, as it provides partial-complete point cloud pairs generated via virtual depth scans, better reflecting real-world LiDAR/RGB-D observations. We further evaluate on ShapeNet-55/34 Wu et al. (2015) and MVP Pan et al. (2021) to assess performance on diverse categories and single-view inputs. Unlike prior diffusion methods that often operate on limited ShapeNet subsets Vahdat et al. (2022); Wei et al. (2025); Kasten et al. (2023); Du et al. (2025), our evaluation follows standardized benchmarks with broader category coverage, providing a more challenging and realistic setting.

We use Chamfer Distance (CD) to evaluate completion quality. CD measures the bidirectional nearest-neighbor distance between predicted and ground-truth point sets, reflecting both geometric coverage and reconstruction errors. As it does not require point ordering or explicit correspondences, CD is well-suited for unordered point clouds. Following prior work, we report both $l1$ and $l2$ CD on different datasets. We also adopt the F-score as an additional evaluation metric Yu et al. (2021).

4.2 Results on PCN dataset

We first report quantitative comparisons on the PCN dataset using $l1$ CD and F1-score. As shown in Tab. 1, our method achieves the best overall $l1$ CD and F1-score on PCN. It also obtains the best or competitive category-level results across the eight categories, indicating improved global structure recovery and local detail reconstruction. We also evaluate AdaPoinTr enhanced with our method, which shows improvement over its original performance.

Qualitative comparisons with SymmCompletion are provided in Appendix E.

4.3 Results on ShapeNet-55/34

For the ShapeNet-55/34 dataset, we also perform a comprehensive quantitative evaluation against SOTA methods. This dataset contains three difficulty subsets. We first train and validate on all 55 categories, then train on 34 categories, and test on both 34 seen categories and 21 unseen categories to evaluate our method's generalisation ability.

Table 3: Quantitative results in terms of $l2$ Chamfer Distance $\times 10^4$ (CD) and F1-Score@1% (F1) on MVP dataset for four densities.

Method	2048		4096		8192		16384	
	CD (↓)	F1 (↑)	CD (↓)	F1 (↑)	CD (↓)	F1(↑)	CD (↓)	F1(↑)
PCNYuan et al. (2018)	9.77	0.32	7.96	0.46	6.99	0.56	6.02	0.64
SnowflakeNet Xiang et al. (2021)	5.71	0.50	4.45	0.65	3.48	0.74	2.69	0.79
PDR Lyu et al. (2022)	5.66	0.49	4.26	0.65	3.35	0.75	2.61	0.82
AEDNet Fu et al. (2024)	5.12	0.52	3.75	0.66	3.21	0.76	2.24	0.83
SymmCompletionYan et al. (2025)	4.89	**0.54**	3.65	0.69	2.70	0.78	2.14	0.85
Ours	**4.85**	0.53	**3.46**	**0.69**	**2.68**	**0.80**	**2.09**	**0.86**

Table 4: Diffusion and conditioning ablations on PCN with 16,384 output points: CD-L1 ($\times 10^3$, ↓). Only the full dual-SUMI model uses SUMI at both Level-1 and Level-2; all other variants use SUMI only at Level-1. Full configurations are given in Appendix B.

Variant	CD-L1 (↓)
Single-SUMI: Level-1 only	
Deterministic refinement (1 step; $T = 0$)	6.33
Iterative deterministic refinement (200 steps)	6.33
Diffusion refinement (200 steps; main model)	6.27
Without coarse/keypoint guidance	7.77
Without partial/symmetry features	6.30
Dual-SUMI: Level-1 and Level-2	
Full dual-SUMI model	6.26

The experimental results are shown in Tab. 2, where CD-S, CD-M and CD-H represent the $l2$ CD at the three difficulty levels. Compared to existing methods, our method achieves strong performance across most settings, with clear improvements on ShapeNet-55 and strong results on both seen and unseen categories. On the more challenging Unseen-21 split, SUMI remains competitive and achieves the best result under the hardest missing setting, suggesting good robustness to unseen categories.

4.4 Results on MVP

We next evaluate completion performance on the MVP dataset at four output densities, ranging from 2,048 to 16,384 points. As shown in Table 3, SUMI achieves the lowest CD among the compared methods at all four densities. Compared with SymmCompletion, CD decreases from 3.65 to 3.46 at 4,096 points, corresponding to a relative reduction of approximately 5.2%, and from 2.14 to 2.09 at 16,384 points, a reduction of approximately 2.3%. The smaller reductions at 2,048 and 8,192 points indicate that the magnitude of the improvement varies across output densities.

For F1-score, SUMI matches the best reported result at 4,096 points and achieves the highest scores at 8,192 and 16,384 points, improving over SymmCompletion from 0.78 to 0.80 and from 0.85 to 0.86, respectively. At 2,048 points, however, the lower CD is accompanied by a slightly lower F1-score (0.53 versus 0.54). Thus, the CD improvements extend across all evaluated densities, while simultaneous improvements in both metrics are observed at the two highest output densities.

4.5 Ablation Study

Model Configuration. We compare single-step deterministic, iterative deterministic, and diffusion refinement using the single-SUMI configuration on PCN with 16,384 output points. Tab. 4 also reports single-SUMI conditioning ablations and the full dual-SUMI variant; all results use CD-L1, where lower is better.

The single-step non-diffusion variant ($T = 0$) retains the refinement backbone and parameter budget and obtains 6.33. We additionally train an iterative deterministic SUMI without diffusion: after the first prediction, each of the remaining 199 steps feeds the preceding prediction back into the network

Table 5: Coarse and dense CD-L1 ($\times 10^3$, $\downarrow$) after end-to-end training. Coarse CD measures LSTNet's direct output P_c before SUMI, rather than the Level-1 refined output.

Pipeline	LSTNet coarse CD	Final dense CD
LSTNet $\rightarrow$ SGFormer	9.81	6.33
LSTNet $\rightarrow$ SUMI $\rightarrow$ SGFormer	**9.70**	6.27
LSTNet $\rightarrow$ SUMI $\rightarrow$ SUMI	9.70	6.26

and updates the state. This 200-step variant also obtains 6.33 at the reported precision, whereas 200-step diffusion refinement achieves 6.27. Thus, increasing deterministic refinement to the same number of steps does not reproduce the diffusion improvement in this configuration, supporting the contribution of diffusion-based refinement beyond iteration count alone.

The conditioning ablations also use SUMI only at Level-1 and are compared with the single-SUMI main model (6.27). Removing coarse/keypoint guidance increases CD-L1 from 6.27 to 7.77; jointly removing partial features F_p and symmetry features F_s, while retaining coarse/keypoint guidance, increases it to 6.30. These results show a larger degradation when coarse/keypoint guidance is removed. Only the full dual-SUMI variant uses SUMI at both Level-1 and Level-2, obtaining 6.26.

Coarse and Dense Predictions under Joint Training. We examine how inserting SUMI at Level-1 affects both the upstream coarse prediction and the final dense reconstruction under end-to-end training. The primary comparison is between the LSTNet–SGFormer baseline and the main configuration, which inserts SUMI between LSTNet and the final SGFormer upsampling stage. In Tab. 5, coarse CD-L1 is evaluated on the direct output P_c of LSTNet, before any SUMI refinement, whereas dense CD-L1 is evaluated after the final upsampling stage.

Although SUMI operates at the Level-1 fine-generation stage, its influence during training can extend to the upstream coarse generator. Since LSTNet is optimized jointly with the refinement and upsampling stages, the refinement losses can also update its parameters. With SUMI at Level-1, the CD-L1 of LSTNet's direct coarse output decreases from 9.81 to 9.70, while final dense CD-L1 decreases from 6.33 to 6.27 with the SGFormer upsampling architecture retained. These results are consistent with end-to-end optimization benefiting both coarse prediction and dense reconstruction. SUMI therefore provides a Level-1 refinement stage while also allowing the upstream coarse generator to benefit from joint training. The comparison does not separately quantify how much of the final improvement comes from the improved coarse prediction and how much comes from the subsequent refinement.

For completeness, we also report the two-stage SUMI variant introduced in the model-configuration ablation. This variant replaces the final SGFormer stage with a second SUMI module and obtains a coarse CD-L1 of 9.70 and a dense CD-L1 of 6.26. The main configuration remains the Level-1-only variant because the second diffusion stage introduces additional computation for a small further reduction in dense CD-L1. Additional ablations and configuration details are in Appendices B–D.

5 Limitation

A limitation of SUMI is that iterative denoising introduces additional inference cost compared with purely feed-forward completion models. To mitigate this, we apply SUMI only at the first fine-generation stage and use a lightweight upsampling module for the final high-density output. Future work may explore accelerated sampling or distillation to further improve efficiency.

6 Conclusion

In this paper, we present a diffusion-enhanced coarse-to-fine framework for PCC. Extensive experiments across multiple benchmarks demonstrate the effectiveness of our approach, achieving competitive or state-of-the-art performance in point cloud completion. Furthermore, SUMI can be integrated into existing coarse-to-fine architectures as a refinement module, showing its potential to enhance future PCC methods.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for polish text and format only. We have not used generative AI tools for all rest tasks. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The architecture and training objectives of SUMI are described in Section 3. Appendix I provides the diffusion noise schedule, timestep sampling strategy, optimization settings, and architectural dimensions. The experimental environment and training costs are reported in Appendix A, while the evaluated model configurations and baseline reproduction protocol are documented in Appendices B and H, respectively. We have implemented the proposed method and verified that the reported performance can be consistently reproduced across repeated runs without selecting a particular random seed. We will publicly release the implementation and the configurations used for training and evaluation to facilitate independent reproduction of our results.

REFERENCES

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9): 10850–10869, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yi Du, Zhipeng Zhao, Shaoshu Su, Sharath Golluri, Haoze Zheng, Runmao Yao, and Chen Wang. Superpc: a single diffusion model for point cloud completion, upsampling, denoising, and colorization. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 16953–16964, 2025.

Fan Duan, Jiahao Yu, and Li Chen. T-corresnet: Template guided 3d point cloud completion with correspondence pooling query generation strategy. In *European conference on computer vision*, pp. 90–106. Springer, 2024.

Zhiheng Fu, Longguang Wang, Lian Xu, Zhiyong Wang, Hamid Laga, Yulan Guo, Farid Boussaid, and Mohammed Bennamoun. Vapcnet: viewpoint-aware 3d point cloud completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12108–12118, 2023.

Zhiheng Fu, Longguang Wang, Lian Xu, Zhiyong Wang, Hamid Laga, Yulan Guo, Farid Boussaid, and Mohammed Bennamoun. Aednet: adaptive embedding and multiview-aware disentanglement for point cloud completion. In *European conference on computer vision*, pp. 127–143. Springer, 2024.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10021–10030, 2023.

Xiaoguang Han, Zhen Li, Haibin Huang, Evangelos Kalogerakis, and Yizhou Yu. High-resolution shape completion using deep neural networks for global structure and local geometry inference. In *Proceedings of the IEEE international conference on computer vision*, pp. 85–93, 2017.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023.

Hexuan Hu, Jianyu Zhang, Tianjin Yang, Qiang Hu, Yufeng Yu, and Qian Huang. Patrans: Pixel-adaptive transformer for edge segmentation of cervical nuclei on small-scale datasets. *Computers in Biology and Medicine*, 168:107823, 2024.

Xiaoliang Ju, Zhaoyang Huang, Yijin Li, Guofeng Zhang, Yu Qiao, and Hongsheng Li. Diffindscene: Diffusion-based high-quality 3d indoor scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4526–4535, 2024.

Yoni Kasten, Ohad Rahamim, and Gal Chechik. Point cloud completion with pretrained text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:12171–12191, 2023.

Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Symmetry descriptors and 3d shape matching. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 115–123, 2004.

Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

Sora Kim, Sungho Suh, and Minsik Lee. Rad: Region-aware diffusion models for image inpainting. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2439–2448, 2025.

Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20943–20953, 2024.

SS Kumar, RS Vinod Kumar, VG Ranjith, S Jeevakala, and S Sajithra Varun. Grey wolf optimized swinunet based transformer framework for liver segmentation from ct images. *Computers and Electrical Engineering*, 117:109248, 2024.

Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12581–12600, 2023.

Tianhong Li, Dina Katabi, and Kaiming He. Return of unconditional generation: A self-supervised representation generation method. *Advances in Neural Information Processing Systems*, 37: 125441–125468, 2024.

Xiangqian Li, Xin Tan, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Point mask transformer for outdoor point cloud semantic segmentation. *Computational Visual Media*, 2025a.

Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer graphics forum*, volume 34, pp. 435–446. Wiley Online Library, 2015.

Yinghui Li, Qianyu Zhou, Jingyu Gong, Ye Zhu, Richard Dazeley, Xinkui Zhao, and Xuequan Lu. Dapointr: Domain adaptive point transformer for point cloud completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 5066–5074, 2025b.

Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. Reconx: Reconstruct any scene from sparse views with video diffusion model. *arXiv preprint arXiv:2408.16767*, 2024.

Hongying Liu, Fuquan Zhang, Yiqing Xu, Junling Wang, Hong Lu, Wei Wei, and Jun Zhu. Tfnet: Transformer-based multi-scale feature fusion forest fire image detection network. *Fire*, 8(2):59, 2025a.

Xiao Liu, Peng Gao, Tao Yu, Fei Wang, and Ru-Yue Yuan. Cswin-unet: Transformer unet with cross-shaped windows for medical image segmentation. *Information Fusion*, 113:102634, 2025b.

Yuzhou Liu, Lingjie Zhu, Hanqiao Ye, Shangfeng Huang, Xiang Gao, Xianwei Zheng, and Shuhan Shen. Bwformer: Building wireframe reconstruction from airborne lidar point cloud with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22215–22224, June 2025c.

Fuchen Long, Ting Yao, Zhaofan Qiu, Lusong Li, and Tao Mei. Pointclustering: Unsupervised point cloud pre-training using transformation invariance in clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21824–21834, 2023.

Dening Lu, Linlin Xu, Jun Zhou, Kyle Yilin Gao, and Jonathan Li. 3dlst: 3d learnable supertoken transformer for lidar point cloud scene segmentation. *International Journal of Applied Earth Observation and Geoinformation*, 140:104572, 2025.

Shitong Luo and Wei Hu. 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.

Changsheng Lv, Mengshi Qi, Xia Li, Zhengyuan Yang, and Huadong Ma. Sgformer: Semantic graph transformer for point cloud-based 3d scene graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4035–4043, 2024a.

Wenyu Lv, Yian Zhao, Qinyao Chang, Kui Huang, Guanzhong Wang, and Yi Liu. Rt-detrv2: Improved baseline with bag-of-freebies for real-time detection transformer, 2024b. URL https://arxiv.org/abs/2407.17140.

Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion, 2022. URL https://arxiv.org/abs/2112.03530.

Omid Nejati Manzari, Hamid Ahmadabadi, Hossein Kashiani, Shahriar B Shokouhi, and Ahmad Ayatollahi. Medvit: a robust vision transformer for generalized medical image classification. *Computers in biology and medicine*, 157:106791, 2023.

Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *ACM Transactions on Graphics (ToG)*, 25(3):560–568, 2006.

Axi Niu, Trung X Pham, Kang Zhang, Jinqiu Sun, Yu Zhu, Qingsen Yan, In So Kweon, and Yanning Zhang. Acdmsr: Accelerated conditional diffusion models for single image super-resolution. *IEEE Transactions on Broadcasting*, 70(2):492–504, 2024.

OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Liang Pan, Tong Wu, Zhongang Cai, Ziwei Liu, Xumin Yu, Yongming Rao, Jiwen Lu, Jie Zhou, Mingye Xu, Xiaoyuan Luo, Kexue Fu, Peng Gao, Manning Wang, Yali Wang, Yu Qiao, Junsheng Zhou, Xin Wen, Peng Xiang, Yu-Shen Liu, Zhizhong Han, Yuanjie Yan, Junyi An, Lifa Zhu, Changwei Lin, Dongrui Liu, Xin Li, Francisco G omez Fern andez, Qinlong Wang, and Yang Yang. Multi-view partial (mvp) point cloud challenge 2021 on completion and registration: Methods and results. *arXiv preprint arXiv:2112.12053*, 2021.

Mark Pauly, Niloy J Mitra, Joachim Giesen, Markus H Gross, and Leonidas J Guibas. Example-based 3d scan completion. In *Symposium on geometry processing*, volume 23, pp. 32. Vienna, 2005.

Ioannis Romanelis, Vlassis Fotis, Athanasios Kalogeras, Christos Alexakos, Adrian Munteanu, and Konstantinos Moustakas. Efficient and scalable point cloud generation with sparse point-voxel diffusion models. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

Yi Rong, Haoran Zhou, Lixin Yuan, Cheng Mei, Jiahao Wang, and Tong Lu. Cra-pcn: Point cloud completion with intra-and inter-level cross-resolution transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 4676–4685, 2024.

Shuyao Shang, Zhengyang Shan, Guangxing Liu, LunQian Wang, XingHua Wang, Zekai Zhang, and Jinglin Zhang. Resdiff: Combining cnn and diffusion model for image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 8975–8983, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. pmlr, 2015.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention, 2021. URL https://arxiv.org/abs/2012.12877.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Haiping Wang, Yuan Liu, Ziwei Liu, Wenping Wang, Zhen Dong, and Bisheng Yang. Vistadream: Sampling multiview consistent images for single-view scene reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26772–26782, 2025a.

Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, and Chunhua Shen. Pointattn: You only need attention for point cloud completion. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 38, pp. 5472–5480, 2024.

Kai Wang, Shaozhang Niu, Qixian Hao, and Jiwei Zhang. Inpdiffusion: Image inpainting localization via conditional diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7771–7779, 2025b.

Peng-Shuai Wang, Chun-Yu Sun, Yang Liu, and Xin Tong. Adaptive o-cnn: A patch-based deep representation of 3d shapes. *ACM Transactions on Graphics (TOG)*, 37(6):1–11, 2018.

Xiaogang Wang, Marcelo H Ang, and Gim Hee Lee. Voxel-based network for shape completion by leveraging edge generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 13189–13198, 2021.

Haodong Wei, Qitao Hu, and Wenhe Liu. Sc-diff: Semantic-collaborative diffusion for multi-class anomaly detection. In *Proceedings of the 9th International Conference on Electronic Information Technology and Computer Engineering*, pp. 935–938, 2025.

Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net++: Point cloud completion by transformer-enhanced multi-step point moving paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):852–867, 2022.

Peixi Wu, Bosong Chai, Hebei Li, Menghua Zheng, Yansong Peng, Zeyu Wang, Xuan Nie, Yueyi Zhang, and Xiaoyan Sun. Spiking point transformer for point cloud classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 21563–21571, 2025a.

Qiuxia Wu, Haiyang Huang, Kunming Su, Zhiyong Wang, and Kun Hu. Dc-pcn: Point cloud completion network with dual-codebook guided quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8441–8449, 2025b.

Yang Wu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Text2lidar: Text-guided lidar point cloud generation via equirectangular transformer. In *European Conference on Computer Vision*, pp. 291–310. Springer, 2024.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Zijie Wu, Yaonan Wang, Mingtao Feng, He Xie, and Ajmal Mian. Sketch and text guided diffusion model for colored point cloud generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8929–8939, 2023.

Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5499–5509, 2021.

Yi Xiao, Qiangqiang Yuan, Kui Jiang, Jiang He, Xianyu Jin, and Liangpei Zhang. Ediffsr: An efficient diffusion probabilistic model for remote sensing image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–14, 2023.

Zeqi Xiao, Wenwei Zhang, Tai Wang, Chen Change Loy, Dahua Lin, and Jiangmiao Pang. Position-guided point cloud panoptic segmentation transformer. *International Journal of Computer Vision*, 133(1):275–290, 2025.

Bojun Xiong, Si-Tong Wei, Xin-Yang Zheng, Yan-Pei Cao, Zhouhui Lian, and Peng-Shuai Wang. Octfusion: Octree-based diffusion models for 3d shape generation. In *Computer Graphics Forum*, volume 44, pp. e70198. Wiley Online Library, 2025.

Hongyu Yan, Zijun Li, Kunming Luo, Li Lu, and Ping Tan. Symmcompletion: High-fidelity and high-consistency point cloud completion with symmetry guidance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9094–9102, 2025.

Long Yang, Qingan Yan, and Chunxia Xiao. Shape-controllable geometry completion for point cloud models. *The Visual Computer*, 33(3):385–398, 2017.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 206–215, 2018.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23174–23184, October 2023.

Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12498–12507, 2021.

Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pp. 728–737. IEEE, 2018.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)*, 42(4):1–16, 2023.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16965–16974, 2024.

Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *European conference on computer vision*, pp. 416–432. Springer, 2022.

Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5826–5835, 2021.

Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14235–14245, June 2023a.

Zhe Zhu, Honghua Chen, Xing He, Weiming Wang, Jing Qin, and Mingqiang Wei. Svdformer: Complementing point cloud via self-view augmentation and self-structure dual-generator. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14508–14518, 2023b.

Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6748–6758, 2023.

SUPPLEMENTARY MATERIAL

A COMPUTATIONAL RESOURCES AND EFFICIENCY

A.1 HARDWARE AND SOFTWARE ENVIRONMENT

All experiments were conducted on a single NVIDIA RTX 4090 GPU. The detailed hardware and software environment is summarized in Tab. A1.

Table A1: Hardware and software environment used in our experiments.

Item	Configuration
GPU	NVIDIA RTX 4090, 24GB memory
CPU	Intel(R) Core(TM) i7-13700K
System memory	64GB
Operating system	Ubuntu 22.04.4 LTS
CUDA version	12.2
Python version	3.7
PyTorch version	1.13.1
Training precision	FP32

A.2 TRAINING COST

We report the per-epoch training time and evaluation time for SUMI on different datasets in Tab. A2. The reported training time includes the full end-to-end optimization of the coarse prediction module, the SUMI refinement module, and the final upsampling module unless otherwise specified.

Table A2: Training cost of SUMI on different datasets.

Dataset	GPU	Batch size	Epochs	Training Mins / Epoch	Evaluation Mins / Epoch
PCN	RTX 4090	16	350	10	20
ShapeNet-55	RTX 4090	16	350	20	140
ShapeNet-34	RTX 4090	16	350	20	50
MVP	RTX 4090	16	40	22	550

B FURTHER DISCUSSION OF MODEL CONFIGURATION

The following discussion supplements the diffusion and conditioning ablations in Tab. 4. Full model configurations are reported in Tab. A3.

Table A3: Model-configuration ablations on PCN with 16,384 output points, measured by CD-L1 ($\times 10^3$, $\downarrow$). Only the full dual-SUMI model uses SUMI at both Level-1 and Level-2; all other variants use SUMI only at Level-1.

Variant	Modification	CD-L1 ($\downarrow$)
Single-SUMI: Level-1 only		
Single-step deterministic SUMI	No diffusion; one prediction ($T = 0$), retaining the refinement backbone and parameter budget	6.33
Iterative deterministic SUMI	No diffusion; trained for iterative refinement, with 200 sequential predictions and feedback of each preceding prediction	6.33
Single-SUMI (main model)	200 diffusion denoising steps at Level-1, followed by SGFormer dense upsampling	6.27
Without coarse/keypoint guidance	Remove coarse/keypoint guidance from Level-1 SUMI; retain diffusion, partial features F_p, and symmetry features F_s	7.77
Without partial/symmetry features	Jointly remove F_p and F_s from Level-1 SUMI; retain diffusion and coarse/keypoint guidance	6.30
Expanded Fusion configuration	Increase the Level-1 SUMI Fusion Blocks from 4 to 8 heads, together with the required increase in feature dimension	**6.18**
Dual-SUMI: Level-1 and Level-2		
Full dual-SUMI model	SUMI at both Level-1 and Level-2, with 4 heads in each Fusion Block	6.26

The new deterministic-iteration control uses a single SUMI module on PCN with a final output of 16,384 points. The deterministic model is trained for iterative refinement without diffusion. Its first network prediction corresponds to the single-step ($T = 0$) setting; the remaining 199 predictions recursively use the preceding prediction as the next input and update the geometric state. Thus, this control performs 200 sequential refinement predictions, rather than repeating a fixed-input prediction. Both the single-step and 200-step deterministic variants obtain CD-L1 of 6.33 at the reported precision, compared with 6.27 for the main 200-step diffusion model. This comparison supports the contribution of diffusion refinement over the tested deterministic iteration strategy at the same step count. This control matches the refinement step count; wall-clock inference time is not evaluated here. These PCN results are separate from the MVP diffusion-timestep study in Appendix C.

Our main configuration applies SUMI only at the Level-1 fine-generation stage and retains the subsequent upsampling module for the final dense output. As an architectural ablation, we also evaluate a two-stage variant that applies SUMI at both Level-1 and Level-2 upsampling. This variant achieves a CD-L1 of 6.26, compared with 6.27 for the main configuration. The result shows that extending SUMI to both fine-generation stages is feasible in the evaluated setting, but yields only a small additional reduction in reconstruction error. Given the additional computational cost of a second diffusion-based refinement stage, we use the Level-1-only configuration for the main experiments.

We also clarify that the noisy feature and diffusion-guided attention are not two independent components in SUMI. The noisy feature is the diffusion state used as the key and value during feature interaction, and the diffusion timestep controls its progressive refinement. Therefore, removing the noisy-feature guidance corresponds to removing the diffusion formulation from the refinement module.

We additionally evaluate a larger Fusion configuration using SUMI only at Level-1. Increasing the Fusion Blocks from 4 to 8 heads improves CD-L1 from the single-SUMI main model's 6.27 to 6.18. However, this variant also requires a larger feature dimension in our implementation and consequently introduces additional parameters. Therefore, the 6.18 result should be interpreted as a capacity-scaling result rather than a strictly parameter-matched ablation. In the main experiments, we retain 4 heads in each Fusion Blocks to remain aligned with the corresponding SymmCompletion

configuration and to provide a fairer comparison without relying on substantially increased model capacity.

The single-SUMI controls support the benefit of diffusion-based refinement beyond simply increasing deterministic refinement steps in the evaluated configuration. The conditioning and Fusion Block capacity ablations also use a single SUMI at Level-1, with 6.27 as their reference. The full dual-SUMI variant separately evaluates adding SUMI at Level-2. Together, these experiments support the diffusion refinement design while distinguishing its iteration control from the conditioning and capacity studies.

C Output Density, AdaPoinTr Insertion Stage, and Diffusion Timesteps

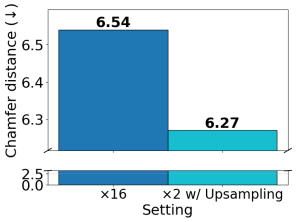


Figure A1: Ablation study on output density for SUMI Module.

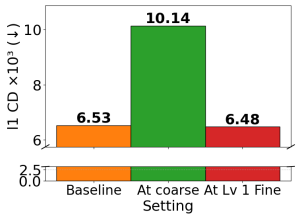


Figure A2: Ablation study on location setting for SUMI Module.

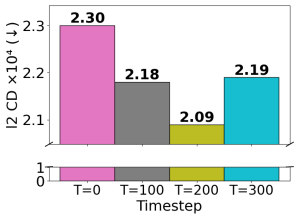


Figure A3: Ablation study on different timesteps for SUMI Module.

We conduct ablation studies on PCN to evaluate the impact of the SUMI Module and diffusion process. Removing the upsampling module and directly generating 16384 points leads to degraded performance (Fig. A1), suggesting that large-scale upsampling hinders fine detail reconstruction. In contrast, generating intermediate 2048-point outputs with SUMI and applying subsequent upsampling preserves more geometric details.

We next examine the insertion stage of SUMI within AdaPoinTr on PCN dataset. As shown in Fig. A2, applying SUMI at the coarse stage significantly degrades performance due to early noise interference, while introducing it at the fine stage yields stable improvements, indicating its effectiveness for local refinement.

We further investigate the effect of diffusion timesteps T on the MVP dataset. As shown in Fig. A3, we evaluate four settings from $T = 0$ to $T = 300$. The $T = 0$ setting keeps the SUMI refinement architecture but disables iterative reverse denoising, serving as a same-architecture single-step refinement baseline. Performance consistently improves as T increases and reaches the best result at $T = 200$, indicating that the gain comes not only from the additional refinement module but also from the diffusion-based iterative denoising process. Increasing the timestep further to $T = 300$ leads to performance degradation, suggesting that $T = 200$ provides a better balance between reconstruction quality and inference efficiency.

D Ablation experiments for diffusion insertion stage

We conducted an additional insertion-stage ablation on our vanilla SUMI pipeline while keeping the network architecture, training settings, and hyperparameters unchanged. We moved the diffusion-based interaction from the refinement stage to the initial coarse-generation stage. The results are reported in Tab.A4, .

Introducing diffusion during the initial coarse-generation stage causes a substantial performance degradation compared with applying it during fine-stage refinement. This result confirms that the conclusion observed with AdaPoinTr also holds for our own base pipeline: diffusion is considerably more effective when used to refine an existing geometric proposal than when introduced before such a proposal has been established.

Table A4: Ablation experiments for diffusion insertion stage. Results are in terms of $l1$ Chamfer Distance $\times 10^3$(CD)

Variant	CD-L1 (↓)
Initial coarse-generation stage	77.6
Fine-stage refinement	6.27

In a coarse-to-fine completion framework, the initial coarse stage is responsible for predicting a geometrically meaningful structure of the missing region from the observed partial point cloud. The observed and missing regions are strongly correlated through their object-level and local geometry. However, the Gaussian noise introduced by diffusion does not initially contain meaningful geometric correspondence with the partial observation. Applying diffusion too early therefore makes the initial structural prediction more ambiguous and unstable. Errors produced at this stage are subsequently propagated to the dense upsampling module, making it more difficult to recover the correct missing geometry.

In contrast, fine-stage diffusion operates on a coarse prediction that already provides an approximate geometric layout. The noisy representation is conditioned on the coarse/keypoint proposal P_c, partial features F_p, and symmetry features F_s during denoising. These geometric constraints guide the reverse process and allow diffusion to refine uncertain structures without discarding the initial relationship between the observed and missing regions.

This confirmatory experiment supports our design choice of placing diffusion in the fine-stage refinement process. The advantage is therefore not simply caused by inserting diffusion anywhere in the pipeline; it depends critically on applying diffusion after a meaningful geometry-conditioned proposal has been generated.

E Qualitative Comparison of Dense Outputs

Fig. A4 presents qualitative comparisons with SymmCompletion. Our method produces more consistent global structures and finer local details, including sharper chair-back slats and smoother surfaces, more accurate airplane wings, improved thin structures (e.g., wires), and better reconstruction of small components like handles and table legs. Overall, our results are closer to the ground truth with fewer artifacts.

F Implementation Setup

We apply the SUMI Module at the Level-1 fine-generation stage to achieve high-quality detail restoration. Its output density is determined by hyperparameters. For example, in the PCN dataset, when the target point cloud density is 16384, the input density of the SUMI Module is set to 1024 and then further improved to the target resolution using a lightweight upsampling module.

We placed the SUMI Module only in the first-stage fine generation, primarily based on the following considerations: First, the DM itself has high computational overhead; reusing it across multiple fine stages would significantly increase training and inference time and may lead to training instability. Second, if the SUMI Module is used only in the second fine stage, while a simple upsampling method is used in the first fine stage, the initial refinement result will still retain errors from the coarse stage, which will be further amplified in subsequent processes, thereby affecting the final completion quality. Therefore, this design achieves a good balance between performance and efficiency.

G Design Rationale and Component Roles

SUMI is a refinement and upsampling module for coarse-to-fine point cloud completion. Its design addresses the interaction between an initial geometric proposal and the subsequent recovery of dense local geometry. An inaccurate intermediate representation leaves the final upsampling stage responsible for recovering both missing structure and the target point density. This motivates refining the intermediate representation before final dense upsampling. The pipeline comparison in

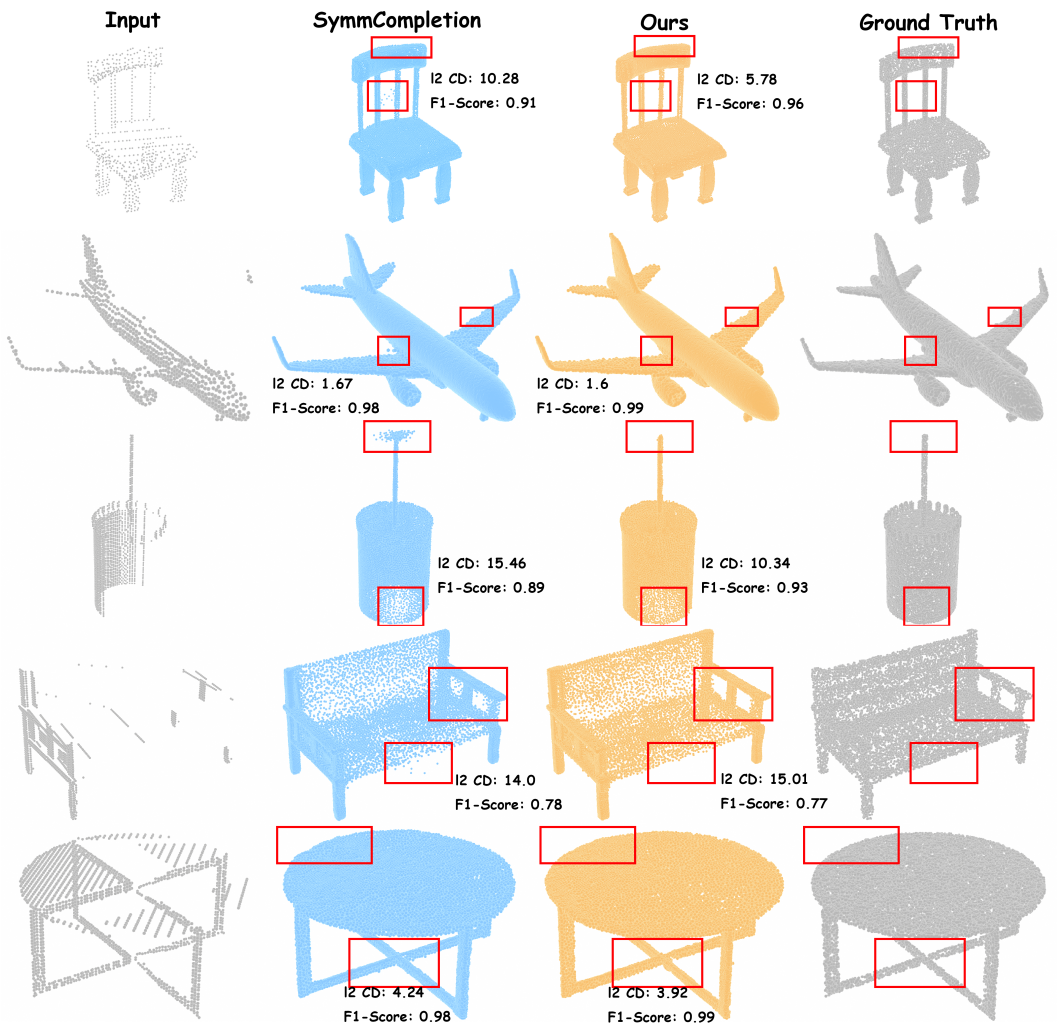


Figure A4: Qualitative comparison between our method and prior works. Our method not only generates high-quality, high-density outputs but also produces fewer local artifacts. We report the $l2$ Chamfer Distance $\times 10^5$ (CD, $\downarrow$) and F1-Score@1% (F1, $\uparrow$) for generation results.

Tab. 5 evaluates this design by reporting LSTNet coarse and dense reconstruction errors with the same downstream upsampling architecture.

Roles of LSTNet, SUMI, and SGFormer. The main configuration follows

$$\text{Partial input} \rightarrow \text{LSTNet} \rightarrow \text{SUMI (Level-1)} \rightarrow \text{SGFormer (Level-2)}.$$

LSTNet, inherited from SymmCompletion Yan et al. (2025), produces the initial coarse prediction and the conditioning features. SUMI uses these features to refine and upsample the prediction at Level-1. The SGFormer architecture is retained for the final dense output. The configuration that applies SUMI at both fine-generation stages is an architectural ablation; the main experiments use the Level-1-only configuration.

Feature interaction and coordinate prediction. SUMI retains attention-based feature interaction while incorporating a noisy point-cloud state into the refinement process. As shown in Fig. 3, the coarse prediction P_c and noisy state x_t are encoded into F_c and F_x^t. The initial cross-attention uses F_c as queries and F_x^t as keys and values. Subsequent branches incorporate partial features F_p and symmetry features F_s, followed by self-attention on the fused representation. The module therefore uses both cross-attention and self-attention. It contains four Fusion Blocks; the default attention configuration uses four heads per attention layer. The eight-head variant in Tab. A3 also increases the feature dimension and is a joint capacity expansion.

The resulting features are decoded into clean point coordinates. Diffusion noise is applied in coordinate space before feature encoding, so the feature-level conditioning does not change the prediction target into a latent feature representation. The contribution concerns the integration of conditional denoising into intermediate refinement within an existing completion pipeline. Comparisons with complete diffusion-based methods also involve differences in backbones, conditioning, and training objectives, and consequently do not isolate this integration strategy as the sole source of performance differences.

H Integration Scope and Baseline Protocol

Integration scope. We evaluate integration with AdaPoinTr and the SymmCompletion-based pipeline. These experiments assess the use of SUMI within two different completion architectures. The refinement module requires a geometric proposal and conditioning features that can be supplied by the host pipeline. Its integration therefore depends on the available features and the interface at the selected refinement stage. The demonstrated scope is the evaluated configurations of these two architectures; compatibility with additional backbones remains to be established. Improving an individual baseline and achieving the best absolute benchmark score are distinct evaluation criteria.

Reported and reproduced baseline results. For the PCN comparison, SymmCompletion Yan et al. (2025) reports a CD-L1 of 6.28, while our reproduction using its official implementation and recommended hyperparameters obtains 6.33. The SymmCompletion result in Tab. 1 is this reproduced result. Under matched data processing, training, and evaluation settings, the main SUMI-augmented configuration obtains 6.27. We distinguish the published score from the reproduced score because the paired evaluation uses the latter as its reference. The paired comparison measures the overall effect of adding SUMI; attribution to individual design choices is addressed separately by the ablations.

I Diffusion Formulation and Training Configuration

I.1 Forward Diffusion and Noise Schedule

For the main configuration, let P_{gt} be the complete ground-truth point cloud. The clean diffusion target is obtained by uniformly downsampling it to 1,024 points:

$$x_0 = \mathcal{U}_{1024}(P_{\text{gt}}), \tag{8}$$

where $\mathcal{U}_{1024}$ denotes uniform point-cloud downsampling. Following DDPM Ho et al. (2020), the forward process perturbs this clean target through Gaussian transitions:

$$q(x_t \mid x_{t-1}) = \mathcal{N}\Big(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I\Big). \tag{9}$$

Let $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, and $\bar{\alpha}_0 = 1$. The marginal distribution at timestep t is

$$q(x_t \mid x_0) = \mathcal{N}\big(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I\big), \tag{10}$$

which permits direct sampling without evaluating all preceding steps:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I). \tag{11}$$

The default configuration uses $T = 200$. During training, each sample's timestep is drawn independently and uniformly from $\{1, \ldots, T\}$. The variance schedule is linear:

$$\begin{aligned} \beta_t &= \beta_1 + \frac{t-1}{T-1}(\beta_T - \beta_1), \\ \beta_1 &= 10^{-4}, \qquad \beta_T = 0.05. \end{aligned} \tag{12}$$

I.2 Variational Background

For completeness, we summarize the standard DDPM variational formulation. This provides background for the reverse process; SUMI's implemented reconstruction objective is specified separately in Appendix I.4. With conditioning suppressed for notational simplicity, the forward and reverse joint distributions factorize as

$$\begin{aligned} q(x_{1:T} \mid x_0) &= \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \\ p_\theta(x_{0:T}) &= p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t), \end{aligned} \tag{13}$$

where the standard terminal prior is $p(x_T) = \mathcal{N}(0, I)$. For a fixed x_0, non-negativity of the KL divergence gives

$$\begin{aligned} \ell_{\mathrm{VLB}}(x_0) &= \mathbb{E}_{q(x_{1:T} \mid x_0)} \left[\log \frac{q(x_{1:T} \mid x_0)}{p_\theta(x_{0:T})} \right] \\ &= -\log p_\theta(x_0) + D_{\mathrm{KL}}(q(x_{1:T} \mid x_0) \,\|\, p_\theta(x_{1:T} \mid x_0)) \\ &\geq -\log p_\theta(x_0). \end{aligned} \tag{14}$$

Using Bayes' rule and collecting the telescoping terms yields

$$\mathcal{L}_{\mathrm{VLB}} = \mathbb{E}_{q(x_0)}[\ell_{\mathrm{VLB}}(x_0)] = \mathbb{E}_q \left[L_T + \sum_{t=2}^{T} L_{t-1} + L_0 \right], \tag{15}$$

where

$$\begin{aligned} L_T &= D_{\mathrm{KL}}(q(x_T \mid x_0) \,\|\, p(x_T)), \\ L_{t-1} &= D_{\mathrm{KL}}(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)), \\ L_0 &= -\log p_\theta(x_0 \mid x_1). \end{aligned} \tag{16}$$

Here the expectation is over the data and forward diffusion process. The decomposition distinguishes the terminal-prior term, the reverse transition terms, and the final reconstruction term.

I.3 Reverse Posterior and Clean-Point Parameterization

For $t \geq 2$, the forward posterior conditioned on the clean point cloud is Gaussian. Applying Bayes' rule gives

$$\begin{aligned} q(x_{t-1} \mid x_t, x_0) &= \frac{q(x_t \mid x_{t-1}) q(x_{t-1} \mid x_0)}{q(x_t \mid x_0)} \\ &= \mathcal{N}\Big(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I\Big), \end{aligned} \tag{17}$$

with mean and variance

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\, x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\, x_0,$$
$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\, \beta_t. \tag{18}$$

Substituting the reparameterization in Eq. 11 gives the equivalent noise-based expression

$$\tilde{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \epsilon \right). \tag{19}$$

These identities motivate either a clean-point or a noise prediction parameterization. In the clean-point parameterization, a network estimate $\hat{x}_{0,t}$ replaces x_0 in the posterior mean:

$$\mu_\theta(x_t, t, C) = \tilde{\mu}_t(x_t, \hat{x}_{0,t}), \tag{20}$$

where C denotes geometric conditioning. For a Gaussian reverse transition with variance $\sigma_t^2 I$, the corresponding sampling form is

$$x_{t-1} = \mu_\theta(x_t, t, C) + \sigma_t z, \qquad z \sim \mathcal{N}(0, I). \tag{21}$$

The posterior distribution in Eq. 17 and a sampled state in Eq. 21 are distinct quantities. The expressions above assume that x_t and $\hat{x}_{0,t}$ have the same dimensionality and coordinate correspondence; $\tilde{\beta}_t$ is the analytic forward posterior variance, whereas σ_t^2 denotes the chosen reverse sampling variance.

The main SUMI configuration uses a 1,024-point reverse state and a 1,024-point clean estimate $\hat{x}_{0,t}$. LSTNet's 1,024-point proposal provides conditioning during this process. The 2,048-point Level-1 output x_f is formed by concatenating the final denoised point cloud P_d with that proposal along the point dimension, as in Eq. 6. This merged output is passed to dense upsampling and is not fed back into the 1,024-point reverse chain.

I.4 SUMI Reconstruction Objective

Given the 1,024-point noisy state x_t and geometric conditions $C = (P_c, F_p, F_s)$, SUMI predicts the clean state estimate $\hat{x}_{0,t}$. The merged Level-1 output x_f contains 2,048 points. Following the main-paper objective, we supervise x_f against the complete ground truth P_{gt}, distinct from the downsampled diffusion target x_0:

$$\mathcal{L}_{\text{diff}} = \mathcal{L}_{CD}(x_f, P_{\text{gt}}). \tag{22}$$

This choice directly supervises completion geometry while allowing the loss to compare unordered point sets. The coarse, Level-1, and final dense outputs are jointly supervised by Eq. 7. SUMI optimizes this geometric reconstruction objective rather than directly minimizing the standard variational bound in Eq. 15.

I.5 Optimization

We use the same default optimizer and learning-rate settings across datasets. AdamW is configured with an initial learning rate of 2×10^{-4} and a weight decay of 5×10^{-4}. The default schedule uses linear warm-up over the first 20 epochs, followed by cosine annealing with a terminal learning rate of 10^{-5} at epoch 350. The actual training durations and batch sizes for each dataset are listed in Tab. A2; the MVP run uses 40 epochs.

I.6 Architectural Dimensions

For the configuration producing 16,384 output points, the main tensor dimensions are listed in Tab. A5. Here B is the batch size. Both the coarse prediction P_c and the noisy diffusion state x_t contain 1,024 points. LSTNet outputs the coarse keypoints P_c with shape $(B, 3, 1024)$ and two distinct feature tensors, partial features F_p and symmetry features F_s, each with shape $(B, 128, 512)$. Keypoint guidance refers to the coarse proposal; F_c denotes its encoded representation. The conditioning feature fusion uses a hidden dimension of 256, and the default attention layers use four

heads. The denoising branch maintains 1,024 points throughout reverse sampling. Its denoised output is concatenated with the 1,024-point LSTNet proposal along the point dimension to produce the 2,048-point Level-1 output x_f. Thus, the 2,048-point output and the 1,024-point reverse state refer to different stages. The merged output is passed to the final upsampling stage to obtain 16,384 points.

Table A5: Tensor dimensions for the main configuration with 16,384 output points. Point-cloud inputs use channel-first layout, while the listed outputs use point-first layout. Conditioning features use batch–channel–point layout.

Tensor	Shape
Coarse keypoints P_c	$(B, 3, 1024)$
Clean diffusion target x_0	$(B, 3, 1024)$
Noisy diffusion state x_t	$(B, 3, 1024)$
Clean estimate $\hat{x}_{0,t}$ (reverse-state layout)	$(B, 3, 1024)$
Partial features F_p	$(B, 128, 512)$
Symmetry features F_s	$(B, 128, 512)$
Merged Level-1 output x_f	$(B, 2048, 3)$
Final dense prediction P_f''	$(B, 16384, 3)$

J Symmetry Guidance and Applicability

Learned local transformations. The symmetry-related prior is provided by the inherited LSTNet coarse module Yan et al. (2025). It predicts local geometric transformations conditioned on the global feature of the partial input. These transformations can vary across local regions, allowing the proposal to adapt to the observed geometry without a predefined global mirror plane. In SUMI, the resulting symmetry features are used together with the coarse/keypoint proposal P_c and partial features F_p as conditioning information for refinement.

Refinement of an imperfect proposal. Symmetry guidance is one component of the conditioning information. The partial-observation features and coarse representation also contribute geometric constraints, and joint supervision of the coarse and refined outputs provides a reconstruction signal for correcting inaccurate proposals. This design permits refinement of an overly symmetric prediction, but does not guarantee recovery when the initial proposal is unreliable. The joint partial/symmetry-feature ablation in Tab. A3 removes both F_p and F_s while retaining coarse/keypoint guidance. It does not isolate symmetry guidance or measure robustness to asymmetric objects.

Applicability limits. Highly asymmetric objects, severe occlusion, and insufficiently observed local structure can produce inaccurate coarse predictions whose errors persist through subsequent refinement. Evaluation on multiple benchmarks and on held-out categories characterizes performance within the reported protocols, but does not establish robustness to severe real sensor noise or to training on one dataset and testing on another. These conditions remain limitations of the current evaluation. Iterative denoising also adds inference cost, motivating the main configuration's use of SUMI only at Level-1.

K Computational resource usage

We further evaluate the VRAM usage for SUMI module. Each SUMI module uses 200 denoising steps; therefore, the SUMI+SGFormer variant contains one iterative diffusion stage, while the full model contains two. We comapred our largest two SUMI modules configuration with SymmCompletion. The results shown in Tab.A6, both our method and SymmCompletion are training and evaluate on a single RTX4090 GPU, with 16 batch size and same point cloud density (1024 $\rightarrow$ 2048 $\rightarrow$ 16384) and 4 Attention-heads in each block.

Our largest model increases peak memory by 1,860 MB, approximately 9.7%, compared with SymmCompletion. The single-SUMI deterministic-iteration comparison in Tab. 4 evaluates refine-

Table A6: Peak GPU memory usage.

Model	Peak GPU memory
SymmCompletion	19,270 MB
Dual-SUMI	21,130 MB

ment quality at the same step count; it is separate from this dual-SUMI memory measurement and does not measure matched inference time.

L NOTATION

For clarity, we summarize the main notation used in the paper in Tab. A7. All symbols are also defined at their first occurrence in the main paper.

Table A7: Summary of notation used in SUMI.

Symbol	Description
P_{gt}	Complete ground-truth reconstruction target
x_0	Clean diffusion target: 1,024 uniformly downsampled GT points
x_t	Noisy point cloud at diffusion timestep t
$\hat{x}_{0,t}$	Clean estimate in the reverse state's coordinate layout
$\tilde{\mu}_t, \tilde{\beta}_t$	Mean and variance of the forward posterior
σ_t	Standard deviation of reverse sampling noise
C	Geometric conditions (P_c, F_p, F_s)
T	Total number of diffusion timesteps
t	Current diffusion timestep
P_c	Coarse keypoint coordinates output by LSTNet
P_d	Final 1,024 denoised points, concatenated with P_c
x_f	Merged Level-1 output: 2,048 points in the main model
P_f	Fine output from the SUMI module
P_f'	Final high-density output after upsampling
F_c	Feature representation encoded from the coarse prediction P_c
F_x^t	Feature representation encoded from the noisy point cloud x_t
F_p	Partial features extracted from the coarse-generation module
F_s	Symmetry features extracted from the coarse-generation module
F_{fusion}	Fused feature after noise-conditioned cross-attention
F_{pf}	Feature obtained by fusing F_{fusion} with partial features
F_{sf}	Feature obtained by fusing F_{fusion} with symmetry features
$\mathcal{C}(\cdot)$	Cross-attention operation
$\mathcal{S}(\cdot)$	Self-attention operation
$\mathcal{L}_{CD}$	Chamfer Distance loss
$\mathcal{L}_{diff}$	Diffusion reconstruction loss
$\mathcal{L}$	Overall training objective