Artist-Created Mesh Generation from Raw Observation

Yao He, Youngjoong Kwon, Wenxiao Cai, Ehsan Adeli
Stanford University

Abstract

We present an end-to-end framework for generating artist-style meshes from noisy or incomplete point clouds, such as those captured by real-world sensors like LiDAR or mobile RGB-D cameras. Artist-created meshes are crucial for commercial graphics pipelines due to their compatibility with animation and texturing tools and their efficiency in rendering. However, existing approaches often assume clean, complete inputs or rely on complex multi-stage pipelines, limiting their applicability in real-world scenarios. To address this, we propose an end-to-end method that refines the input point cloud and directly produces high-quality, artist-style meshes. At the core of our approach is a novel reformulation of 3D point cloud refinement as a 2D inpainting task, enabling the use of powerful generative models. Preliminary results on the ShapeNet dataset demonstrate the promise of our framework in producing clean, complete meshes.

1. Introduction

Artist-created meshes (AMs) play a central role in 3D applications due to their compatibility with commercial graphics pipelines. These meshes offer high-quality topology using relatively few vertices, making them ideal for efficient rendering and editing. However, generating artist-style meshes from real-world sensor data remains a significant challenge.

Existing mesh generation pipelines [5, 6, 11, 20, 35] often assume access to clean and complete 3D inputs, such as dense point clouds or high-fidelity scans. In practice, however, point clouds captured by real-world sensors—like LiDAR or mobile RGB-D cameras—are frequently sparse, noisy, or incomplete. Moreover, conventional approaches rely on complex, multi-stage processing pipelines, which are difficult to scale and prone to cascading errors.

To address these limitations, we propose an end-to-end framework that generates high-quality artist-style meshes directly from raw point cloud inputs, even when the input data is noisy or partial. At the core of our method is a novel formulation of 3D point cloud refinement as a 2D inpainting problem, enabling the use of powerful 2D generative models (*i.e.*, Stable Diffusion [19]).

Specifically, we first project the input 3D point cloud onto a spherical atlas using sphere offsetting followed by equirectangular projection. This produces a 2D point cloud atlas that retains geometric and structural information in image space. We then apply a denoising U-Net to inpaint the atlas, filling in missing regions and reducing noise. The refined atlas is subsequently mapped back to 3D and converted into a clean, complete point cloud. Finally, we pass this point cloud through a frozen feed-forward transformer [6] to generate the final artist-style mesh.

Our contribution can be summarized as follows:

- An end-to-end framework for generating artist-style meshes directly from raw point cloud inputs.
- A novel formulation of 3D point cloud refinement as a 2D inpainting problem.
- Promising results on the ShapeNet dataset [3], showing the potential of our approach for real-world applications.

2. Related Work

2.1. Point Cloud Refinement

The point cloud refinement task aims to recover missing geometry from partial or noisy point clouds. Early deep learning approaches [9, 10, 22, 23] leveraged 3D convolutional networks on voxelized inputs, but these were constrained by high computational and memory costs. PointNet [17, 18] introduced a paradigm shift by directly operating on unordered point sets, enabling more scalable processing. Building on this, PCN [32] and subsequent works [14, 24, 26, 29] adopted a coarse-to-fine folding-based architecture to generate dense point clouds in an end-to-end fashion, conditioned on partial input. Transformer-based methods have since improved completion quality. However, they often suffer from limited contextual understanding, especially when relying solely on 3D input. To address this, recent methods incorporate image guidance. For instance, MBVN [13] employs a conditional GAN to complete missing regions in 2D space and reconstructs 3D point clouds from the completed images. ViPC [34] and Cross-PCC [28] use both the partial point cloud and a single-view image for completion. SVDFormer [36] and PCDreamer [25] take a further step by synthesizing novel views and using them to guide point cloud generation. In particular, PCDreamer utilizes a diffusion-based prior to generate realistic multi-view images.

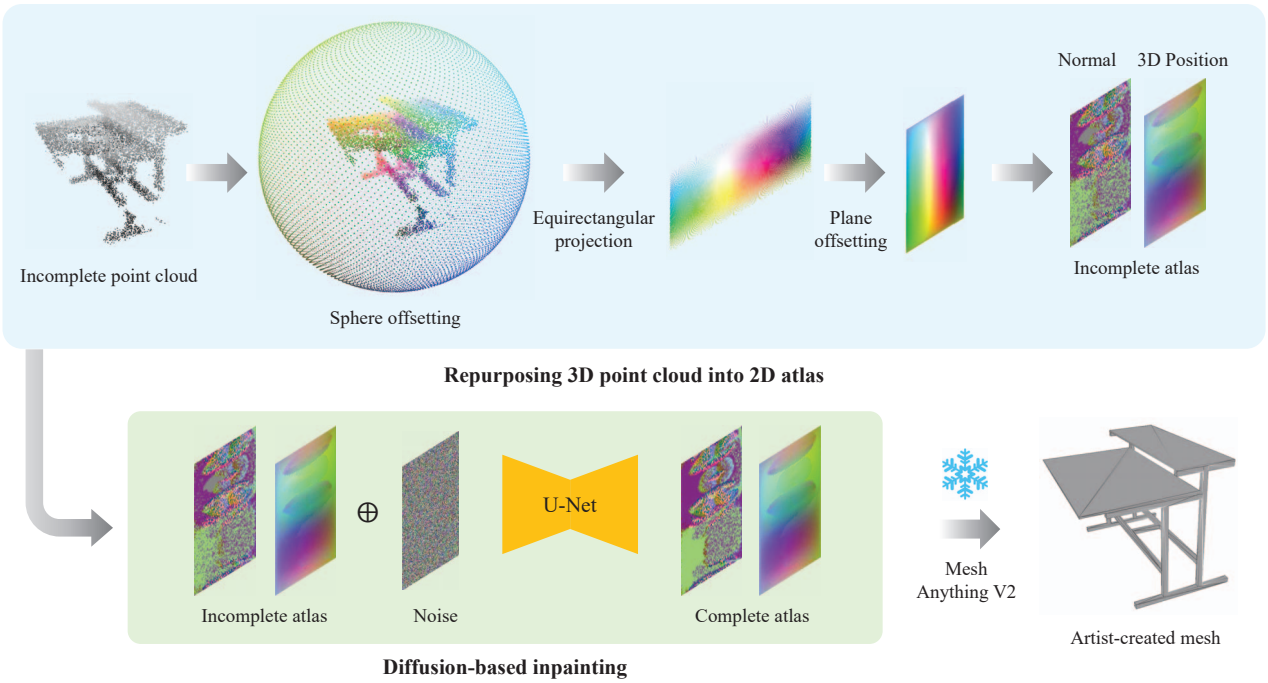

Figure 1. **Overview of Our Pipeline.** *Top*: Representing 3D Point Cloud as 2D Atlas. For each fitting, the point cloud is first translated to the surface of a standard sphere S via optimal transport, where points with the same color indicate a 1-to-1 mapping. Then, the surface points are flattened onto the 2D plane via equirectangular projection with reusable indices. We obtain Atlas by reorganizing the flattened coordinates to pixels of a dense 2D square of size $\sqrt{N} \times \sqrt{N}$. *Bottom*: Training And Inference Pipeline. We fine-tune a latent diffusion model to inpaint incomplete atlases. The completed atlas is then used to reconstruct the full point cloud through inverse mapping of optimal transport. The point cloud is subsequently passed to an off-the-shelf mesh generation model to produce the final artist-created mesh.

However, due to the lack of strong multi-view consistency in diffusion models, the resulting 3D reconstructions often remain noisy. Importantly, most prior works focus solely on completing the missing points without regard for mesh topology. As a result, meshing these outputs often leads to overly complex and unstructured surfaces. Our approach not only refubes the point cloud but also explicitly aims to produce artist-created meshes with clean topology, suitable for downstream use in commercial graphics pipelines.

2.2. Artist-created Mesh Generation

Recent works on artist-created mesh generation focus on producing meshes that efficiently and faithfully capture the underlying topology. A prominent line of research [1, 4, 5, 16, 21, 27] formulates mesh generation as a sequence modeling task, where the mesh is represented as an ordered sequence of faces or vertices. PolyGen [16] employs two autoregressive transformer models to generate vertex coordinates and face indices separately. MeshXL [4] directly autoregresses over sequences of explicit vertex coordinates to generate raw meshes. MeshGPT [20] tokenizes geometric structures into a learned vocabulary and decodes these tokens into faces using an autoregressive model. Polydiff [1] represents the mesh as a quantized triangle soup and applies a diffusion model to denoise it into a clean, coherent mesh. To improve structure and quality, PivotMesh [27] introduces pivot vertices as coarse anchors to guide mesh token generation. MeshAnything [6] leverages a VQ-VAE to learn a compact mesh vocabulary and generates meshes conditioned on input shapes using an autoregressive transformer. Its successor, MeshAnythingV2 [7], improves tokenization efficiency by encoding each face with a single vertex. Meshtron [12] proposes a scalable solution using an hourglass-style transformer architecture for mesh sequence generation. These approaches aim to bridge the gap between raw geometric reconstructions and structured, artist-quality outputs suitable for commercial use. In our work, we align with this goal by generating artist-created meshes from incomplete or noisy point clouds, while also ensuring topological consistency and mesh simplicity through a 2D inpainting-driven design.

3. Method

We formalize the task of generating a high-fidelity artist-created mesh from an incomplete and noisy point cloud. Let $\mathcal{P}_{\text{gt}} \subset \mathbb{R}^3$ denote the ground truth point cloud and $\mathcal{P}_i \subset \mathcal{P}_{\text{gt}}$ denote a subset of the ground truth. Our input is a noisy

corrupted version of $\hat{\mathcal{P}}$, defined as:

$$\hat{\mathcal{P}} = \left\{ p + n_p \mid p \in \mathcal{C}_i(\mathcal{P}_{\text{gt}}),\ n_p \sim \mathcal{N}(0, \sigma^2 I) \right\}. \quad (1)$$

where $\mathcal{C}_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a corruption operator on a point cloud, selecting a subset of $\mathcal{P}_{\text{gt}}$ and $\mathcal{N}_i$ represents the noise applied to the points. The resulting point cloud captures real-world scanning artifacts such as sparsity, occlusions, and sensor noise. Our desired output is a mesh $\boldsymbol{M}$ adopting the form of ordered sequence of faces, vertices and coordinates as described in [11]. We adopt this representation to align with the standard sequence generation formulation for mesh generation. However, due to the noisy and corrupted nature of the input point cloud, additional steps are necessary to address these challenges.

We aim to learning the distribution $P(M \mid \hat{S})$ where M represents the AM and $\hat{S}$ represents the 3D shape condition from the incomplete point cloud $\hat{P}$. In contrast, prior work [5, 6, 11, 35] focuses on learning the distribution $P(M \mid S)$ where S is the 3D shape condition from the complete and noise-free point clouds. A straightforward approach is to fine-tune these models so that they adapt from $P(M \mid S)$ to $P(M \mid \hat{S})$. However, as pointed out in [11], these methods often fail to produce meaningful results when conditioned on incomplete or noisy 3D data. To address this challenge, we instead factorize the distribution as:

$$P(M \mid \hat{S}) = P(M \mid S)\, P(S \mid \hat{S}) \quad (2)$$

This factorization suggests a two-stage approach: first, recover the complete shape condition from its corrupted counterpart by modeling $P(S \mid \hat{S})$, then, generate the AM using the standard pipeline based on $P(M \mid S)$. An overview of our method is demonstrated in Fig. 1.

3.1. Recovering Complete Shapes

Our key intuition is to leverage the powerful distribution modeling capabilities of diffusion models to recover the complete shape condition S from the corrupted version $\hat{S}$, i.e., learning $P(S \mid \hat{S})$. While diffusion models have demonstrated strong generative performance in the 2D domain, 3D diffusion model has been limited by the scarcity of high-quality 3D data, resulting in performance gaps relative to their 2D counterparts. To overcome this limitation, we adopt recent approaches that re-purpose pre-trained 2D diffusion models for 3D object generation through atlasing [30, 31, 33]. Our pipeline for representing 3D point clouds as 2D atlas is shown on top of Fig. 1.

Specifically, this re-purposing process begins by hypothesizing a unit sphere centered around the object, populated with N 3D points $\boldsymbol{s} = \{s_i | s_i \in \mathbb{R}^3\}$ uniformly distributed on its surface. Each point in the input point cloud is then mapped onto the sphere surface via Optimal Transport (O. T.) [2]. Once positioned, an equirectangular projection is applied to convert the spherical coordinates to flattened 2D coordinates $m_i \in \mathbb{R}^2$. A second O. T. step maps these projected coordinates m_i onto the vertices $n_i \in \mathbb{R}^2$ of a regular $\sqrt{N} \times \sqrt{N}$ 2D grid. While prior methods apply Gaussian atlases to encode features such as 3D location, color, opacity, scale, and rotation, our point cloud setting includes only 3D locations and normals. Therefore, the final output is an atlas $\mathbf{X} \in \mathbb{R}^{\sqrt{N} \times \sqrt{N} \times (||\boldsymbol{x}-\boldsymbol{s}||+||\boldsymbol{n}||)}$ where $\boldsymbol{x} - \boldsymbol{s}$ and $\boldsymbol{n}$ denote the offset 3D coordinates and normals, respectively.

Upon obtaining the atlas map, we fine-tune the pre-trained Latent Diffusion (*i.e.*, Stable Diffusion [19]) to inpaint missing areas of the given incomplete atlas. Following [30], our inpaint U-Net $\mathcal{U}$ directly operates on the atlas without involving VAE. At each diffusion step t, the inpaint network $\mathcal{U}$ predicts the noise ϵ added to the ground-truth atlas $\mathbf{X}$, conditioned on the input incomplete atlas $\hat{\mathbf{X}}$. Specifically, we condition the generation on the incomplete atlas by adding it to the noisy image (*i.e.*, $\mathbf{X}_t + \hat{\mathbf{X}}$). Although we used the simple conditioning for the initial exploration, in the future work, more sophisticated conditioning (*e.g.*, self-attention) can be employed. The training objective minimizes the difference between the predicted and the ground-truth noise: $\mathcal{L} = \mathbb{E}\left[\|\epsilon - \mathcal{U}(\mathbf{X}_t + \hat{\mathbf{X}}, i)\|^2\right]$, where $\mathbf{X}_t = \alpha_t \mathbf{X} + \sigma_t \epsilon$. Here, $\mathbf{X}$ is the ground-truth latent, $\epsilon \sim \mathcal{N}(0, I)$ represents Gaussian noise, and α_i and σ_i are diffusion parameters that define the noise level at diffusion timestep i.

During inference time, we use $T = 20$ denoising steps to inpaint the corrupted atlas. The completed atlas is then transformed back into a 3D point cloud representation from the bijective O. T. mapping, thereby closing the loop for learning the distribution $P(S \mid \hat{S})$.

3.2. Recovering Meshes

Given an estimate of $P(S \mid \hat{S})$, we can approximate $P(M \mid \hat{S})$ by first computing $P(M \mid S)$ as described in Eq. 2. At this stage, the availability of a complete point cloud allows us to employ an off-the-shelf mesh generation model. Specifically, we use MeshAnything V2 [5] to estimate $P(M \mid S)$. It formulates mesh generation as an auto-regressive token prediction problem on a learned mesh vocabulary. Given a fully completed shape S condition, i.e., the encoded point cloud, it infers the conditional distribution

$$P(M \mid S) = \prod_{t=1}^{T} P\big(m_t \mid m_{1:t-1},\, S\big) \quad (3)$$

where each token m_t represents either a vertex coordinate quantized to a fixed grid or a face index triplet.

4. Experiments

In this section, we benchmark and evaluate the effectiveness of our proposed method on the test set from our dataset described above. MeshAnything V2 [5] serves as our primary

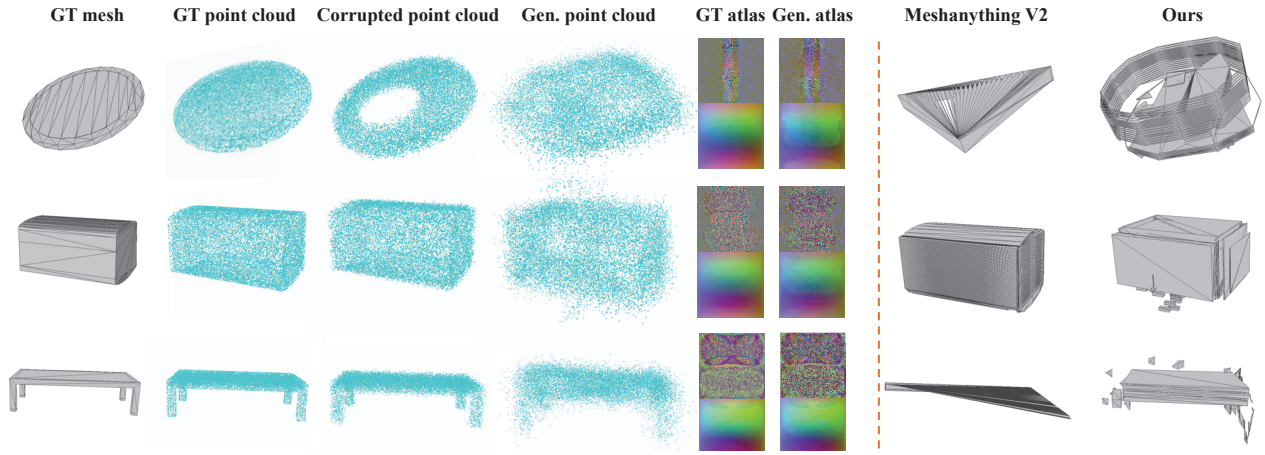

Figure 2. **Qualitative Results.** We compare the meshes generated by our method and MeshAnything V2. The figure also includes the ground-truth mesh, point cloud sampled from the ground-truth meshes, corrupted input point cloud, ground-truth Atlas maps, and their generated counterparts. The results demonstrate that our approach can reconstruct key geometric features where MeshAnything V2 fails, while also highlighting several limitations of our model.

Table 1. Comparison of our method and MeshAnything V2 on the noisy and cropped ShapeNet dataset.

Method	CD↓ ($\times 10^{-2}$)	ECD↓ ($\times 10^{-2}$)	NC↑	#V↓	#F↓	V_Ratio↓	F_Ratio↓
Ours	0.913	**1.283**	0.344	**381.2**	**635.0**	**3.505**	**1.599**
MeshAnything V2	**0.909**	1.422	**0.393**	527.4	998.1	4.705	2.398

baseline. To quantitatively assess mesh quality, we adopt the following standard metrics setting as in [8]: Chamfer Distance (CD), Edge Chamfer Distance (ECD), Normal Consistency (NC), Vertex and Face Counts ($\#V$, $\#F$), Vertex/Face Ratios (V_Ratio, F_Ratio).

4.1. Results & Discussion

We present the evaluation results of our method and MeshAnything V2 in Table 1. Fig. 2 shows qualitative examples of the generated meshes.

Overall, the evaluation results in Table 1 show that our approach achieves performance comparable to, and in some cases surpassing, baseline, demonstrating its effectiveness under noisy point cloud input conditions. In particular, our method achieves a similar CD, and a better ECD, suggesting comparable mesh quality. However, it performs worse in NC, indicating challenges in accurately reconstructing surface normals. This may be attributed to the standard diffusion process not enforcing unit-length constraints, leading to noisier outputs. This insight suggests a future improvement in the training pipeline.

We observe that our method achieves lower values in the metrics $\#V$, $\#F$, V_Ratio, F_Ratio compared to MeshAnything V2. While lower values in these metrics typically indicate higher ability to represent complex topology using fewer vertices and faces, this does not apply in our case. A sanity check of the generated results reveals that many generated meshes exhibit missing parts. Thus, the reduced values more likely reflect a deficiency in preserving geometric completeness. We provide further hypotheses for this behavior in the supplementary materials.

5. Conclusions

We presented an end-to-end framework for generating artist-style meshes from noisy or incomplete point clouds, addressing key limitations of existing methods that assume clean inputs or rely on multi-stage pipelines. By reformulating the 3D point cloud refinement problem as 2D inpainting, our approach leverages powerful 2D generative models to recover clean and complete geometry. Preliminary results demonstrate the potential of our method to produce high-quality meshes from real-world-like inputs.

Acknowledgment

This work was partially supported by Panasonic Holdings Corporation, UST, and the Stanford Human-Centered AI (HAI) Google Cloud Credits.

References

[1] Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417*, 2023. 2

[2] Rainer E Burkard and Eranda Cela. Linear assignment problems and extensions. In *Handbook of combinatorial optimization: Supplement volume A*, pages 75–149. Springer, 1999. 3

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[4] Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Billzb Wang, Jingyi Yu, Gang Yu, et al. Meshxl: Neural coordinate field for generative 3d foundation models. *Advances in Neural Information Processing Systems*, 37:97141–97166, 2024. 2

[5] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 1, 2, 3

[6] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. *arXiv preprint arXiv:2406.10163*, 2024. 1, 2, 3

[7] Yiwen Chen, Yikai Wang, Yihao Luo, Zhengyi Wang, Zilong Chen, Jun Zhu, Chi Zhang, and Guosheng Lin. Meshanything v2: Artist-created mesh generation with adjacent mesh tokenization. *arXiv preprint arXiv:2408.02555*, 2024. 2, 1

[8] Zhiqin Chen, Andrea Tagliasacchi, Thomas Funkhouser, and Hao Zhang. Neural dual contouring. *ACM Transactions on Graphics (Special Issue of SIGGRAPH)*, 41(4), 2022. 4

[9] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017. 1

[10] 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*, pages 85–93, 2017. 1

[11] Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024. 1, 3

[12] Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. Meshtron: High-fidelity, artist-like 3d mesh generation at scale. *arXiv preprint arXiv:2412.09548*, 2024. 2

[13] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4completion: Synthesizing multi-view depth maps for 3d shape completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[14] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11596–11603, 2020. 1

[15] Carsten Moenning and Neil A Dodgson. Fast marching farthest point sampling. Technical report, University of Cambridge, Computer Laboratory, 2003. 1

[16] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pages 7220–7229. PMLR, 2020. 2

[17] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[18] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 3

[20] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19615–19625, 2024. 1, 2

[21] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19615–19625, 2024. 2

[22] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1955–1964, 2018. 1

[23] Jacob Varley, Chad DeChant, Adam Richardson, Joaquín Ruales, and Peter Allen. Shape completion enabled robotic grasping. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 2442–2447. IEEE, 2017. 1

[24] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 790–799, 2020. 1

[25] Guangshun Wei, Yuan Feng, Long Ma, Chen Wang, Yuanfeng Zhou, and Changjian Li. Pcdreamer: Point cloud completion through multi-view diffusion priors. *arXiv preprint arXiv:2411.19036*, 2024. 1

[26] 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. 1

[27] Haohan Weng, Yikai Wang, Tong Zhang, CL Chen, and Jun Zhu. Pivotmesh: Generic 3d mesh generation via pivot vertices guidance. *arXiv preprint arXiv:2405.16890*, 2024. 2

[28] Lintai Wu, Qijian Zhang, Junhui Hou, and Yong Xu. Leveraging single-view images for unsupervised 3d point cloud completion. *IEEE Transactions on Multimedia*, 2023. 1

[29] 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*, pages 5499–5509, 2021. 1

[30] Tiange Xiang, Kai Li, Chengjiang Long, Christian Häne, Peihong Guo, Scott Delp, Ehsan Adeli, and Li Fei-Fei. Repurposing 2d diffusion models with gaussian atlas for 3d generation. *arXiv preprint arXiv:2503.15877*, 2025. 3, 1

[31] Haitao Yang, Yuan Dong, Hanwen Jiang, Dejia Xu, Georgios Pavlakos, and Qixing Huang. Atlas gaussians diffusion for 3d generation. *arXiv preprint arXiv:2408.13055*, 2024. 3

[32] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 international conference on 3D vision (3DV)*, pages 728–737. IEEE, 2018. 1

[33] Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. Gaussiancube: A structured and explicit radiance representation for 3d generative modeling. *arXiv preprint arXiv:2403.19655*, 2024. 3

[34] Xuancheng Zhang, Yutong Feng, Siqi Li, Changqing Zou, Hai Wan, Xibin Zhao, Yandong Guo, and Yue Gao. View-guided point cloud completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15890–15899, 2021. 1

[35] Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. *arXiv preprint arXiv:2503.15265*, 2025. 1, 3

[36] 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*, pages 14508–14518, 2023. 1