MeshXL: Neural Coordinate Field for Generative 3D Foundation Models

Sijin Chen[1,2,*] **Xin Chen**[1,†] **Anqi Pang**[1] **Xianfang Zeng**[1] **Yijun Fu**[1]
Wei Cheng[1] **Fukun Yin**[1,2] **Yanru Wang**[1] **Zhibin Wang**[1]
Jingyi Yu[3] **Gang Yu**[1] **Bin Fu**[1] **Tao Chen**[2,‡]

[1]Tencent PCG [2]Fudan University [3]ShanghaiTech University
[†] project lead [‡] corresponding author

Figure 1: **We present MeshXL, a family of generative pre-trained transformers, for the direct generation of 3D meshes.** We validate that Neural Coordinate Field, an explicit coordinate representation with implicit neural embeddings, is a simple-yet-effective sequence representation for large-scale mesh modelling.

Abstract

The polygon mesh representation of 3D data exhibits great flexibility, fast rendering speed, and storage efficiency, which is widely preferred in various applications. However, given its unstructured graph representation, the direct generation of high-fidelity 3D meshes is challenging. Fortunately, with a pre-defined ordering strategy, 3D meshes can be represented as sequences, and the generation process can be seamlessly treated as an auto-regressive problem. In this paper, we validate the **Neur**al **C**oordinate **F**ield (NeurCF), an explicit coordinate representation with implicit neural embeddings, is a simple-yet-effective representation for large-scale sequential mesh modelling. After that, we present MeshXL, a family of generative pre-trained auto-regressive models, which addresses the process of 3D mesh generation with modern large language model approaches. Extensive

[*]Research done when Sijin Chen was a Research Intern at Tencent PCG.

Preprint. Under review.

experiments show that MeshXL is able to generate high-quality 3D meshes, and can also serve as foundation models for various down-stream applications.

1 Introduction

The generation of high-quality 3D assets [61, 79, 29] is essential for various applications in video games, virtual reality, and robotics. Among existing 3D representations [51, 38, 57, 61], the 3D mesh represents the 3D data with graphs, which has the flexibility and accuracy for sharp edges as well as both flat and curved surfaces. However, the direct generation of high-quality 3D meshes is challenging, given 1) the unstructured graph representation and 2) the demand for accurate spatial locations and connectivity estimation within vertices.

To generate 3D meshes, many works adopt an indirect way by first producing data in other 3D representations, including point clouds [99, 49, 54], SDF [90, 96], and multi-view images [46, 84, 30]. After that, re-meshing methods [37] are required for post-processing the generated geometries. There are also attempts towards the direct generation of 3D polynomial meshes. PolyGen [53] adopts two separate decoder-only transformers for vertices generation and connectivity prediction. MeshGPT [66] builds a mesh VQVAE to reconstruct the tokens generated by a GPT model [59] into 3D meshes. Meanwhile, PolyDiff [2] directly adopts discrete denoising diffusion [4] on the discretized mesh coordinates.

Though these methods have achieved initial success in 3D assets generation, they suffer from certain limitations. To preserve high-frequency information, the point cloud and voxel representations will make dense samplings on the object surfaces, which inevitably lead to great redundancy when representing flat surfaces. The reconstruction-based methods [84, 30, 68], however, rely heavily on the quality of the multi-vew generation pipeline [46]. Additionally, the VQVAE-based 3D generation methods [90, 66] will inevitably result in cumulative errors when reconstructing the generated tokens into 3D structures.

To tackle the above challenges and explore the potential of scaling up 3D generative pre-training, we first introduce a simple-yet-effective way of 3D mesh representation, the **Neur**al **C**oordinate **F**ield (NeurCF). NeurCF represents the explicit 3D coordinates with implicit neural embeddings. We show that with a pre-defined ordering strategy, the generation of 3D meshes can be formulated as an auto-regressive problem. After that, we present MeshXL, a family of generative pre-trained transformers [95, 59], for the direct generation of high-fidelity 3D meshes. Through NeurCF, we can train large-scale 3D models to generate 3D meshes in an end-to-end manner, which simplifies both the mesh preparation and generation pipeline.

Extensive experiments demonstrate that representing 3D meshes with NeurCF facilitates MeshXL to generate higher-quality 3D meshes with an increased number of parameters and large-scale pre-training data. By training on the collection of large-scale 3D mesh data, MeshXL can achieve better performance with larger numbers of parameters (Fig. 3 and Tab. 4), and surpass prior arts on multiple categories task of the ShapeNet dataset [9] (Tab. 2).

In summary, our contributions can be summarized as follows:

- We validate that Neural Coordinate Field is a simple-and-effective representation of 3D mesh, which is also friendly to large-scale auto-regressive pre-training.
- We present a family of MeshXLs that can be treated as strong base models for image-conditioned or text-conditioned 3D mesh generation tasks.
- We show that MeshXL surpasses state-of-the-art 3D mesh generation methods, and can produce delicate 3D meshes compatible with existing texturing methods.

2 Related Work

First, we present a concise review of existing 3D representations. Subsequently, we discuss related works on 3D generation and recent efforts in developing 3D foundation models.

3D Representations. Researchers have long sought for accurate and efficient methods to represent 3D data. **Point Cloud** [54, 57, 58, 91] captures the spatial positions of discrete points in the Euclidean

space, which is preferred by various 3D sensors [15, 89, 67, 3, 7]. **Mesh** [53, 2, 66, 12] represents the 3D structure with graphs. By connecting the vertices with edges, mesh can also be interpreted into a set of polygons in the 3D space. Similar to point clouds, **3D Gaussians** [38, 69] also record the discrete Euclidean distribution in 3D space. However, each point is represented by a 3D Gaussian distribution function parameterized by its covariance matrix, color, and opacity. Given their fast convergence and rendering speed, 3D gaussians are often utilized for 3D reconstruction. **Neural Radiance Field** (NeRF) [51, 5] constructs a learnable volumetric function f using neural networks trained on multi-view images. Due to its derivability and flexibility, NeRF is also favored for 3D generative models [46, 101, 78, 56]. Additionally, there are other 3D representations such as multi-view images [76, 92, 102], voxel fields [61, 13, 45], and signed distance fields [96], among others [65, 90, 64]. In this paper, we consider the **Neural Coordinate Field** (NeurCF), an explicit spatial representation with implicit neural embeddings, and investigate its potential for scalable 3D asset generation.

3D Generation. With the exploration of various 3D representations and the collection of large-scale 3D datasets [17, 9, 16], researchers have also put much effort exploring the generation of high-fidelity 3D assets [42, 39]. The **G**enerative **A**dversarial **N**etwork (GAN) [25, 82, 1, 33] produces synthetic 3D data with a generator $\mathcal{G}$, and train a discriminator network $\mathcal{D}$ to distinguish the generated and real data. Additionally, the potential of **diffusion** models [54, 28, 62] in the direct generation of 3D data is also widely explored [99, 2, 54, 50, 47]. The key idea behind diffusion is to transform the desired data distribution into a simpler distribution (*e.g.* gaussian) and learn a desnoising model for the reverse process. Besides, researchers have also explored the potential of diffusion models in generating **multi-view** images [46, 16, 84, 43], and reconstruct them into 3D structures. In this paper, we mainly explore the **auto-regressive** methods for 3D generation. AutoSDF [52] and MeshGPT [66] learn to generate discrete tokens and reconstruct them into 3D representations with a VQVAE model [73]. PolyGen [53] adopts two decoder-only transformers that predict the location and connectivity of vertices, sequentially. In this paper, we explore the potential of an explicit sequential modelling method for 3D meshes, and present a family of generative pre-trained transformers, MeshXL, for high-fidelity 3D mesh generation.

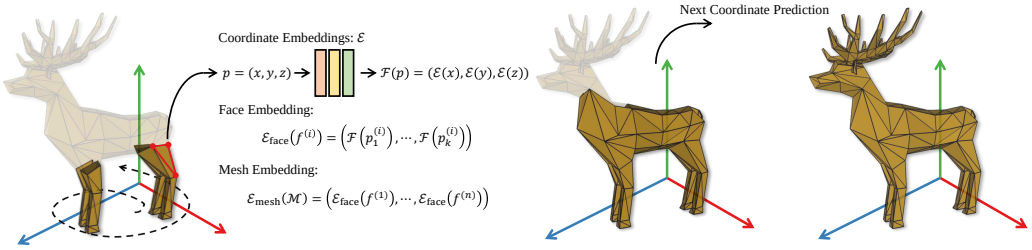
3D Foundation Models. The collection of large-scale high-quality 3D data [17, 16, 9, 83, 72, 21, 22] builds up the foundation for various 3D-related tasks [85, 27, 10, 41]. To explore the scaling effects in 3D learning, researchers have made great endeavors in building 3D foundation models for 3D understanding [98, 44, 100, 87, 88, 94, 102], reconstruction [30, 80, 68, 46, 16, 86, 75], and generation [61, 29, 66, 8]. With the introduction of large-scale 3D data in both variety and granularity [34, 41, 16], existing 3D foundation models are capable of generalizing to unseen concepts [102, 88, 44], generating high-fidelity 3D assets [90, 36, 66], responding to complex instructions [31, 10, 32, 41], and generating actions that interacts with the 3D environments [20, 81, 97]. In this paper, we present a fully end-to-end 3D mesh generation pipeline, explore the scaling effect for large-scale pre-training, and test whether our method can serve as a well-trained foundation model for various down-stream tasks.

3 Neural Coordinate Field

Neural **C**oordinate **F**ield (NeurCF) is an explicit representation with implicit neural embeddings. To be specific, for a Euclidean 3D coordinate system, we can partition the vertices coordinates into an N^3 grid. Then, each discretized coordinate $p = (x, y, z)$ can be encoded with the coordinate embedding layer $\mathcal{E}$, where $\mathcal{F}(p) = (\mathcal{E}(x), \mathcal{E}(y), \mathcal{E}(z))$. Therefore, a k-sided polynomial face $f^{(i)}$ can be encoded with $\mathcal{E}_{\text{face}}(f^{(i)}) = (\mathcal{F}(p_1^{(i)}), \cdots, \mathcal{F}(p_k^{(i)}))$. For simplicity, the learnable coordinate embeddings $\mathcal{E}$ are shared among axes.

Ordering. Due to the graph representation, the order of the mesh vertices and the order of the edges between them are permutation-invariant. A pre-defined ordering strategy is essential to facilitate the sequence modelling in MeshXL. We employ the same ordering strategy as PolyGen [53] and MeshGPT [66]. The mesh coordinates are first normalized into a unit cube based on the mesh's longest axis, and discretized into unsigned integers. Within each face, the vertices are cyclically permuted based their coordinates (z-y-x order, from lower to higher), which helps to preserve the direction of normal vectors. Then, we order these faces based on the permuted coordinates (lower to

high). To this end, an n-faced 3D k-sided polynomial mesh can be represented as $\mathcal{M} \in \mathbb{Z}^{n \times k \times 3}$, and we can encode $\mathcal{M}$ with $\mathcal{E}_{\text{mesh}} = (\mathcal{E}_{\text{face}}(f^{(1)}), \cdots, \mathcal{E}_{\text{face}}(f^{(n)}))$.



(a) Neural Coordinate Field (b) Auto-regressive Mesh Generation

Figure 2: **Mesh Representation.** We present the **Neur**al **C**oordinate **F**ield (NeurCF) to encode the discretized coordinates in the Euclidean space. Benefiting from NeurCF and a pre-defined ordering strategy, our proposed MeshXL can directly generate the unstructured 3D mesh auto-regressively.

A Sequential Mesh Representation. One direct way to represent the 3D meshes is to directly reshape $\mathcal{M}$ into a vector with $(n \cdot k \cdot 3)$ tokens. As a special case, an n-faced triangular mesh can be represented by a vector with $9n$ tokens. Meanwhile, our representation can also be expanded to hybrid polynomial mesh representations with the proper introduction of separate tokens. For example, we can generate triangles within "<tri> $\cdots$ </tri>" and quadrilaterals within "<quad> $\cdots$ </quad>". To identify the start and end of a mesh sequence, we add a <bos> ("begin-of-sequence") token before the mesh sequence and an <eos> ("end-of-sequence") token after.

Comparisons. Compared to other forms of 3D representations, NeurCF is a direct representation for 3D meshes. Since we represent each coordinate with learnable embeddings, NeurCF is an end-to-end trainable representation for unstructured 3D meshes. Additionally, NeurCF is storage efficient comparing to voxel fields ($O(N^3)$) and point clouds, since it can naturally model the flat surfaces with graph structures.

4 Method

In this section, we present the architecture and training objectives for MeshXL models. Following this, we investigate the effects of scaling.

Architecture. In Sec. 3, we present a simple-yet-effective way to represent a 3D mesh into a sequence. Therefore, the learning of 3D mesh generation can be formulated into an auto-regressive problem, and can be seaminglessly addressed by modern **L**arge **L**anguage **M**odel (LLM) approaches. In our paper, we adopt the decoder-only transformers using the OPT [95] codebase as our base models. To adapt the pre-trained OPT models to our *next-coordinate prediction* setting, we fine-tune the whole model with newly-initialized coordinate and position embeddings.

Generative Pre-Training. We use the standard next-token prediction loss to train our models. Given the trainable weights θ and an $|s|$-length sequence s, the generation loss is calculated as:

$$\mathcal{L}_{\text{MeshXL}}(\theta) = -\sum_{i=1}^{|s|} \log P\left(s_{[i]} | s_{[1,\cdots,i-1]}; \theta\right). \tag{1}$$

For each mesh sequence, we add a <bos> token before the mesh tokens, and an <eos> token after the mesh tokens to identify the ending of a 3D mesh. During inference, we adopt the top-k and top-p sampling strategy to produce diverse outputs.

Conditional Mesh Generation. Here we mainly consider generating 3D meshes from images and texts. For image to 3D generation, we extract 2D features with a frozen ViT [19] model. Likewise, we extract text features with a frozen BERT [18] model. The overall training objective of the conditional mesh generation is shown in Eq. (2).

$$\mathcal{L}_{\mathcal{X}\text{-to-mesh}}(\theta) = -\sum_{i=1}^{|s|} \log P\left(s_{[i]} | s_{[1,\cdots,i-1]}; \mathcal{X}\right) \tag{2}$$

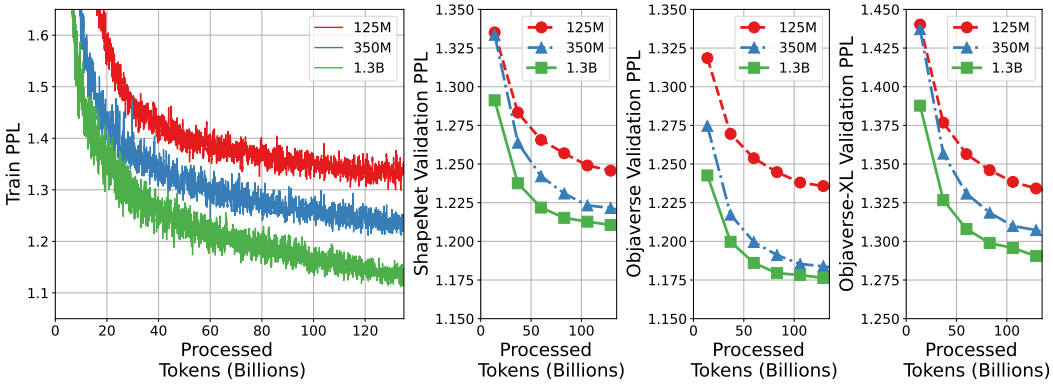


Figure 3: **Training and Validation Perplexity (PPL) for MeshXL Models.** We train all the models from scratch on 150 billion tokens. We observe that the performance grows with model sizes.

Scaling Up. We present MeshXL in various sizes, including 125M, 350M, and 1.3B. The detailed hyperparameters for training different models can be found in Tab. 1. To better analyze the scaling effects, we train all models from scratch on 150 billion tokens. We provide both training curve and validation perplexity for different models in Fig. 3. One can see that as the number of parameters grows, the model achieves a lower validation perplexity, indicating a higher probability to produce the validation data.

5 Experiments

We first briefly introduce the data, metrics, and implementation details in Sec. 5.1. Then, we provide evaluations and comparisons on the generated meshes (cf. Sec. 5.2) and ablations (cf. Sec. 5.3). We also provide visualization results in Sec. 5.4.

5.1 Data, Metrics, and Implementation Details

Data. We pre-train the base model with 2.5 million 3D meshes collected from the combination of ShapeNet [9], 3D-FUTURE [22], Objaverse [17], and Objaverse-XL [16]. We use planar decimation on meshes with more than 800 faces following MeshGPT [66] and RobustLowPoly [11]. More details on the data collection and processing pipeline can be found in the appendix. For generative mesh pre-training, we randomly rotate these meshes with degrees from (0°, 90°, 180°, 270°), and adopt random scaling along each axis within range $[0.9, 1.1]$ for data augmentation.

Metrics. We follow the standard evaluation protocols in MeshGPT [66] and PolyDiff [2] with the following metrics. Coverage (COV) is sensitive to mode dropping and is used to quantify the diversity of the generated meshes. However, COV does not assess the quality of the generated results. Minimum Matching Distance (MMD) calculates the average distance between the reference set and their closest neighbors in the generated set. However, MMD is not sensitive to low-quality results. The 1-Nearest Neighbor Accuracy (1-NNA) directly quantifies the quality and diversity between the generation set and the reference set. The optimal value of 1-NNA is 50%. We adopt the Jensen-Shannon Divergence (JSD) score to directly evaluate 3D meshes. We use Chamfer Distance to measure the similarity between two samples. We also adopt the Frechet Inception Distance (FID) and Kernel Inception Distance (KID) on the rendered images for feature-level evaluation. The MMD, JSD, and KID scores are multiplied by 10^3.

Implementation. All experiments are conducted on a cluster consisting of 128 A100 GPUs. We train our models under bfloat16 and the ZeRO-2 strategy [60] using the AdamW [48] optimizer with a learning rate decaying from 10^{-4} to 10^{-6} and a weight decay of 0.1. The detailed hyperparameters for different models can be found in Tab. 1. To train our base models, we load the weights from the pre-trained OPT models [95] and initialize the word embeddings and positional embeddings from scratch. Without further specification, we generate 3D meshes with the top-k and top-p sampling strategy with $k = 50$ and $p = 0.95$.

Table 1: **Hyperparameters for different MeshXL Base Models.** We present three MeshXL models with 125M, 350M, and 1.3B parameters, respectively.

Hyperparameters	MeshXL(125M)	MeshXL(350M)	MeshXL(1.3B)
# Layers	12	24	24
# Heads	12	16	32
d_{model}	768	1,024	2,048
d_{FFN}	3,072	4,096	8,192
Optimizer	AdamW(β_1=0.9, β_2=0.999)		
Learning rate	1.0×10^{-4}	1.0×10^{-4}	1.0×10^{-4}
LR scheduler	Cosine	Cosine	Cosine
Weight decay	0.1	0.1	0.1
Gradient Clip	1.0	1.0	1.0
Number of GPUs	8	16	32
# GPU hrs (A100)	1,944	6,000	23,232

5.2 Evaluations and Comparisons

We provide quantitative as well as qualitative comparisons on both unconditional and conditional 3D mesh generation on public benchmarks.

Unconditional Generation. We evaluate our methods as well as several baseline methods using the ShapeNet [9] data in Tab. 2. For each category, we split the data by 9:1 for training and validation. To produce and evaluate category specific 3D meshes, we fine-tune our pre-trained base model with a global batch size of 16 on each category for about 100k iterations, and sample 1,000 meshes for each category. As the code for MeshGPT [66] is not available, we mainly compare our methods with PolyGen [53] and GET3D [23].

As can be seen from Tab. 2, PolyGen achieves a significantly lower COV scores on chair and table and a high 1-NNA score. This indicates that PolyGen has low generative diversity and produces results close to the training data. Meanwhile, our proposed MeshXL families demonstrate superiority over PolyGen and GET3D in both COV and MMD. Furthermore, the MeshXL families achieve 1-NNA scores closer to 50%, indicating the great quality and diversity of the generated samples. To sum up, MeshXL can not only produces high quality samples, but also preserves diversity.

Table 2: **Quantitative Comparisons with Prior Arts.** To produce category-specified 3D meshes, we fine-tune the pre-trained MeshXL models on the ShapeNet [9] subsets. Among the listed metrics, we scale MMD, JSD, KID by 10^3.

Category	Methods	COV↑	MMD↓	1-NNA	JSD↓	FID↓	KID↓
Chair	PolyGen [53]	7.79	16.00	99.16	228.80	63.49	43.73
	GET3D [23]	11.70	15.92	99.75	155.25	67.84	42.10
	MeshXL (125M)	50.80	**3.11**	56.55	9.69	28.15	1.48
	MeshXL (350M)	50.80	3.17	**55.80**	9.66	28.29	**1.39**
	MeshXL (1.3B)	**51.60**	3.23	**55.80**	**9.48**	**9.12**	1.84
Table	PolyGen [53]	44.00	3.36	67.20	25.06	54.08	14.96
	GET3D [23]	16.80	10.39	91.90	226.97	67.65	34.62
	MeshXL (125M)	51.21	2.96	57.96	**12.82**	42.55	**0.92**
	MeshXL (350M)	49.70	3.07	**56.10**	13.64	43.43	1.27
	MeshXL (1.3B)	**52.12**	**2.92**	56.80	14.93	**22.29**	2.03
Bench	PolyGen [53]	31.15	4.01	83.23	55.25	70.53	0.012
	MeshXL (125M)	54.37	1.65	**43.75**	16.43	**35.31**	**0.82**
	MeshXL (350M)	53.37	1.65	42.96	**15.41**	36.35	0.96
	MeshXL (1.3B)	**56.55**	**1.62**	39.78	15.51	35.50	1.60
Lamp	PolyGen [53]	35.04	7.87	75.49	96.57	65.15	12.78
	MeshXL (125M)	**55.86**	5.06	48.24	43.41	34.61	**0.84**
	MeshXL (350M)	53.52	**4.18**	**49.41**	**34.87**	**25.94**	1.92
	MeshXL (1.3B)	51.95	4.89	47.27	41.89	31.66	0.99

Conditional Generation. For conditional generation, we adhere to the data pre-processing pipeline in A.2 to obtain text-mesh and image-mesh pairs for training and sampling. To make MeshXL

understand the additional condition, we compress the text/image feature with the Q-Former [40] to align the text/image feature with mesh coordinate embeddings.

User Study. To evaluate how well the generated 3D meshes align with human preference, we perform user studies on the chair category in Tab. 3 with several baseline methods [53, 23]. We recruit and instruct the participants to score each mesh from 0 to 5 based on its 1) **quality**: the smoothness of object surfaces and completeness of the mesh, 2) **artistic**: how much do you believe this object is designed and created by artists, and 3) **triangulation**: how well do the connectivity among vertices aligns with the models created by professional designing software [14]. For the above mentioned metrics, the higher score means better quality. As a baseline evaluation, we also ask the participants to score the ground truth 3D geometries sampled from the ShapeNet data. We have collected a total of 434 valid responses, and the results show that the 3D meshes created by MeshXL are consistently preferred by human in all dimensions.

Table 3: **User Study.** Compared to baseline methods, the meshes generated by MeshXL are better aligned with human preference in terms of both geometry and designs.

Methods	Quality↑	Artistic↑	Triangulation↑
PolyGen [53]	2.53	2.72	3.15
GET3D [23]	3.15	2.46	3.15
MeshXL	**3.96**	**3.45**	**3.72**
Ground Truth	4.08	3.33	3.75

5.3 Ablation Studies

Effectiveness of Model Sizes. To analyze whether a larger model pre-trained on the collection of large-scale 3D mesh data benefits 3D mesh generation, we evaluate MeshXL base models with different sizes on the Objaverse [17] dataset in Tab. 4. We observe that as the model size grows, the generated samples exhibits a closer 1-NNA to 50%, a larger COV, and smaller JSD score, which indicates an improving diversity and quality.

Table 4: **Effectiveness of Model Sizes.** We observe that as the model size grows, the generated meshes exhibit a closer 1-NNA to 50%, a larger COV and a smaller JSD, indicating better diversity and quality.

Method	COV↑	MMD↓	1-NNA	JSD↓	FID↓	KID ↓
MeshXL (125M)	39.76	5.21	67.34	26.03	17.32	4.48
MeshXL (350M)	40.79	5.21	65.68	23.71	**15.14**	**3.35**
MeshXL (1.3B)	**41.73**	**5.10**	**64.53**	**21.86**	15.50	3.71

Shape Completion. To analysis whether our method is capable of producing diverse outputs, we ask MeshXL (1.3B) model to predict the whole object given some partial observations of the 3D mesh. In practice, we use 50% of the object mesh as input, and ask the model to predict the rest 50% of the object. We illustrate completion examples on chairs and tables in Fig. 4. One can see that Mesh-XL is able to produce diverse outputs given the partial observation of the 3D mesh.

5.4 Visualizations

We provide qualitative comparisons on the meshes generated by our method as well as the meshes generated by other baseline models.

Random Sampling. We visualize 3D meshes randomly sampled from MeshXL base model in Fig. 6. After training on a large-scale collection of 3D mesh data, MeshXL is able to produce diverse and high-quality 3D meshes.

Qualitative Comparison. We provide category specified visualization results as well as their normal vectors on the generated meshes in Fig. 7. With the ability to generate 3D meshes directly, MeshXL is able to produce high-quality 3D meshes with both sharp edges and smooth surfaces.

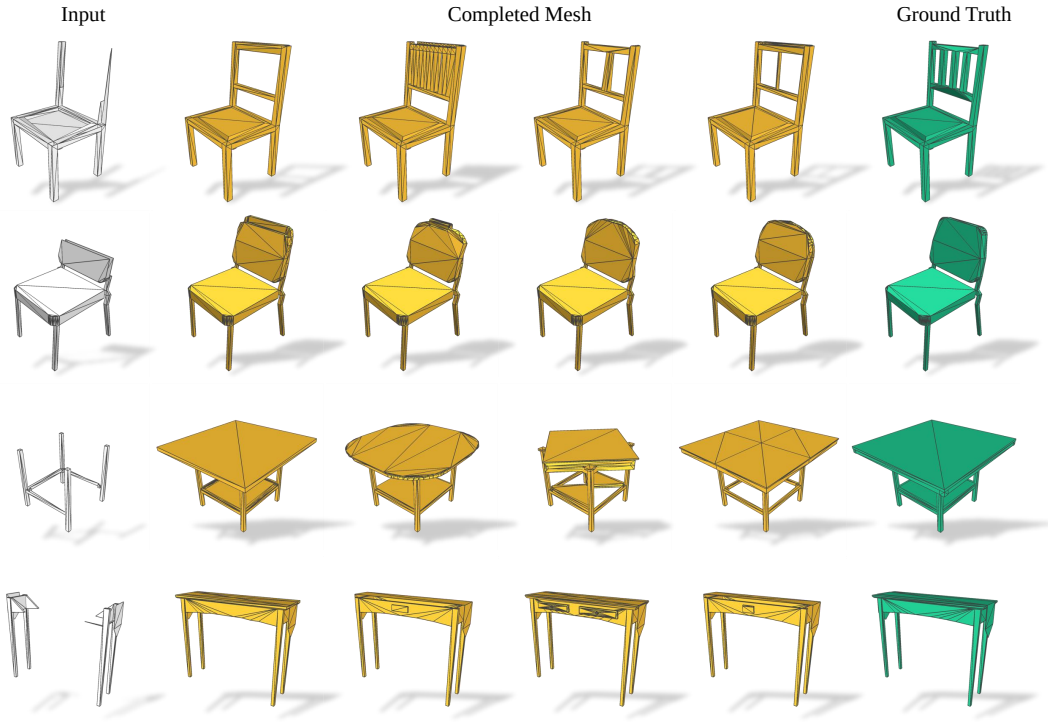


Figure 4: **Evaluation of Partial Mesh Completion.** Given some partial observation of the 3D mesh (gray), MeshXL is able to produce diverse object completion results (yellow).

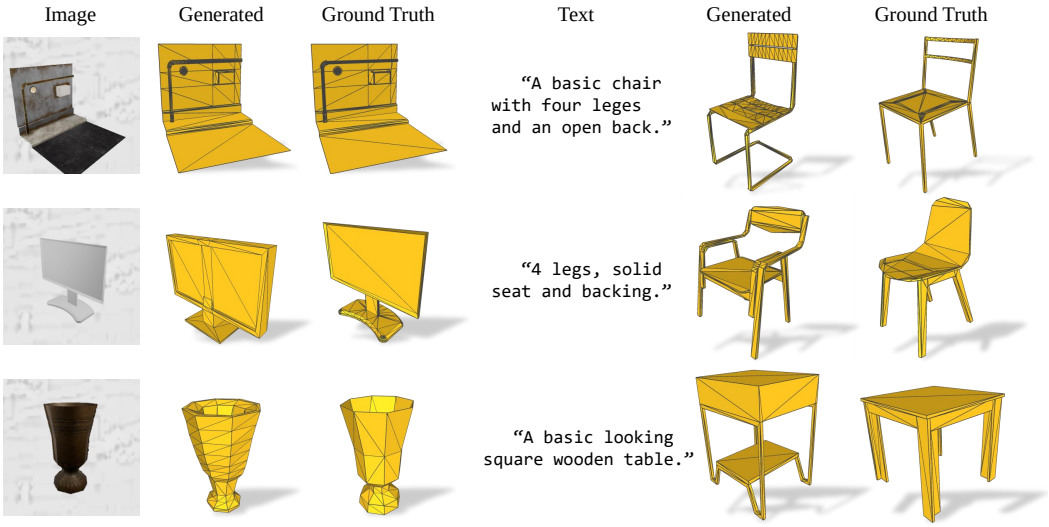


Figure 5: **Evaluation of conditional mesh generation.** We show that MeshXL can generate high-quality 3D meshes given the corresponding image or text as the additional inputs.

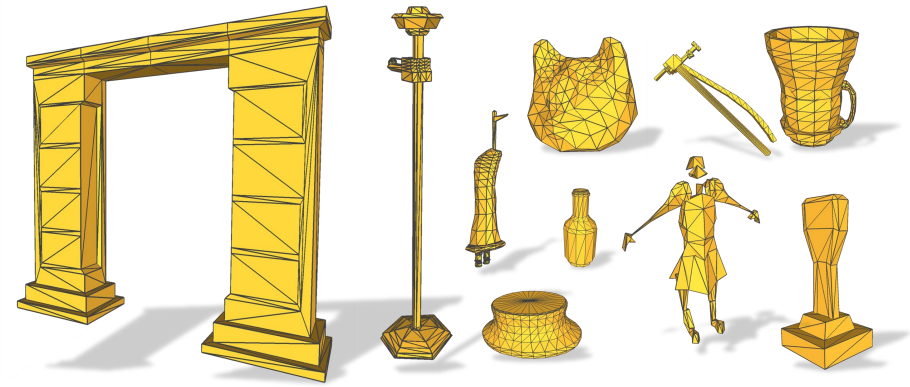


Figure 6: **Gallery of random samples from MeshXLs.** After training on a large-scale collection of 3D mesh data, MeshXL is able to produce diverse and high-quality 3D meshes.

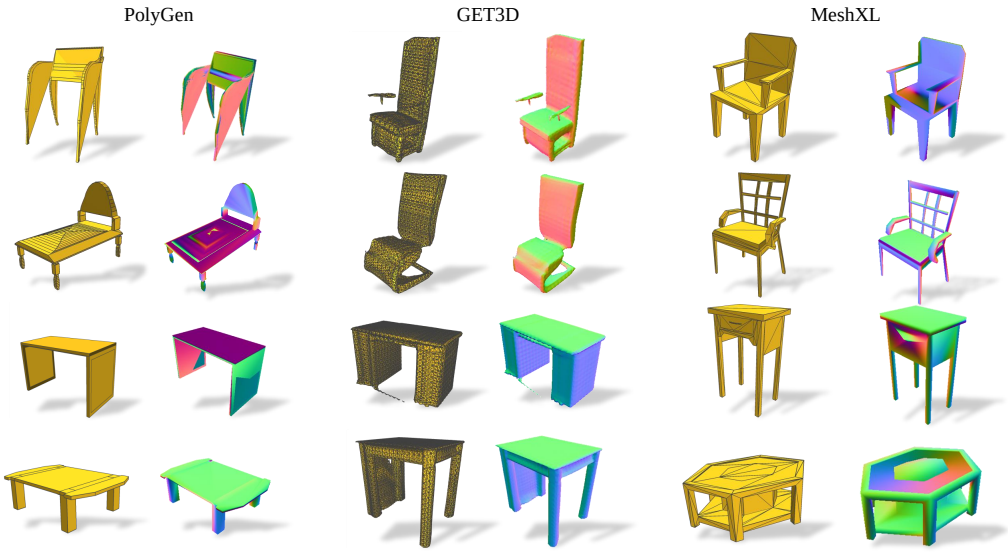


Figure 7: **Qualitative Comparison on the Generated Meshes.** We present qualitative comparisons on the generated meshes as well as normal vectors. MeshXL is able to produce high-quality 3D meshes with both sharp edges and smooth surfaces.

6 Limitations, Future Work, and Conclusion

Limitations and Future Work. The main drawback of MeshXLs is the inference time. During sampling, MeshXL will generate 7,200 tokens for an 800-faced 3D mesh, which takes a relatively long time because of the auto-regressive process. As for future works, recent endeavors on the RNN-related methods [6, 55, 26] and multiple tokens prediction for LLMs [24] might open up great opportunities in saving the inference cost.

Conclusion. We validate that NeurCF, an explicit coordinate representation with implicit neural embeddings, is a simple-and-effective representation of 3D meshes. By modelling the 3D mesh generation as an auto-regressive problem, we seek help from modern LLM approaches and present a family of generative pre-trained models, MeshXL, for high-fidelity 3D mesh generation. We show that MeshXL performs better given larger-scale training data and increased parameters. Extensive results show our proposed MeshXL can not only generate high-quality 3D meshes, but also exhibits great potential serving as base models for conditional 3D assets generation.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.

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

[3] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016.

[4] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[5] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021.

[6] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024.

[7] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.

[8] Ziang Cao, Fangzhou Hong, Tong Wu, Liang Pan, and Ziwei Liu. Difftf++: 3d-aware diffusion transformer for large-vocabulary 3d generation. *arXiv preprint arXiv:2405.08055*, 2024.

[9] 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.

[10] Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding, reasoning, and planning. *arXiv preprint arXiv:2311.18651*, 2023.

[11] Zhen Chen, Zherong Pan, Kui Wu, Etienne Vouga, and Xifeng Gao. Robust low-poly meshing for general 3d models. *ACM Transactions on Graphics (TOG)*, 42(4):1–20, 2023.

[12] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 45–54, 2020.

[13] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[14] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[16] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[17] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[20] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[21] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021.

[22] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, pages 1–25, 2021.

[23] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022.

[24] Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. Better & faster large language models via multi-token prediction. *arXiv preprint arXiv:2404.19737*, 2024.

[25] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[26] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[27] Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, et al. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*, 2023.

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

[29] Fangzhou Hong, Jiaxiang Tang, Ziang Cao, Min Shi, Tong Wu, Zhaoxi Chen, Tengfei Wang, Liang Pan, Dahua Lin, and Ziwei Liu. 3dtopia: Large text-to-3d generation model with hybrid diffusion priors. *arXiv preprint arXiv:2403.02234*, 2024.

[30] 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.

[31] Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

[32] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[33] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13559–13568, 2021.

[34] Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. *arXiv preprint arXiv:2401.09340*, 2024.

[35] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[36] Biao Jiang, Xin Chen, Wen Liu, Jingyi Yu, Gang Yu, and Tao Chen. Motiongpt: Human motion as a foreign language. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[38] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[39] Chenghao Li, Chaoning Zhang, Atish Waghwase, Lik-Hang Lee, Francois Rameau, Yang Yang, Sung-Ho Bae, and Choong Seon Hong. Generative ai meets 3d: A survey on text-to-3d in aigc era. *arXiv preprint arXiv:2305.06131*, 2023.

[40] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[41] Mingsheng Li, Xin Chen, Chi Zhang, Sijin Chen, Hongyuan Zhu, Fukun Yin, Gang Yu, and Tao Chen. M3dbench: Let's instruct large models with multi-modal 3d prompts. *arXiv preprint arXiv:2312.10763*, 2023.

[42] Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024.

[43] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023.

[44] Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. Openshape: Scaling up 3d shape representation towards open-world understanding. *Advances in Neural Information Processing Systems*, 36, 2024.

[45] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[46] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023.

[47] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023.

[48] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[49] 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*, pages 2837–2845, 2021.

[50] Zhaoyang Lyu, Jinyi Wang, Yuwei An, Ya Zhang, Dahua Lin, and Bo Dai. Controllable mesh generation through sparse latent point diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 271–280, 2023.

[51] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[52] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022.

[53] 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.

[54] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[55] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

[56] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

[57] 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.

[58] 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.

[59] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[60] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020.

[61] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube ($\mathcal{X}^3$): Large-scale 3d generative modeling using sparse voxel hierarchies. *arXiv preprint arXiv:2312.03806*, 2023.

[62] 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.

[63] 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.

[64] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

[65] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023.

[66] 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. *arXiv preprint arXiv:2311.15475*, 2023.

[67] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[68] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[69] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

[70] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[71] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[72] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.

[73] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[74] 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.

[75] Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023.

[76] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, et al. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. *arXiv preprint arXiv:2312.16170*, 2023.

[77] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[78] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[79] Zhenwei Wang, Tengfei Wang, Gerhard Hancke, Ziwei Liu, and Rynson W.H. Lau. Themestation: Generating theme-aware 3d assets from few exemplars. 2024.

[80] Xinyue Wei, Kai Zhang, Sai Bi, Hao Tan, Fujun Luan, Valentin Deschaintre, Kalyan Sunkavalli, Hao Su, and Zexiang Xu. Meshlrm: Large reconstruction model for high-quality mesh. *arXiv preprint arXiv:2404.12385*, 2024.

[81] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023.

[82] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016.

[83] 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*, pages 1912–1920, 2015.

[84] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

[85] Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. *arXiv preprint arXiv:2308.16911*, 2023.

[86] Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024.

[87] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1179–1189, 2023.

[88] Le Xue, Ning Yu, Shu Zhang, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. *arXiv preprint arXiv:2305.08275*, 2023.

[89] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22, 2023.

[90] Fukun Yin, Xin Chen, Chi Zhang, Biao Jiang, Zibo Zhao, Jiayuan Fan, Gang Yu, Taihao Li, and Tao Chen. Shapegpt: 3d shape generation with a unified multi-modal language model. *arXiv preprint arXiv:2311.17618*, 2023.

[91] Wang Yu, Xuelin Qian, Jingyang Huo, Tiejun Huang, Bo Zhao, and Yanwei Fu. Pushing the limits of 3d shape generation at scale. *arXiv preprint arXiv:2306.11510*, 2023.

[92] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9150–9161, 2023.

[93] Xianfang Zeng. Paint3d: Paint anything 3d with lighting-less texture diffusion models. *arXiv preprint arXiv:2312.13913*, 2023.

[94] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8552–8562, 2022.

[95] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[96] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024.

[97] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

[98] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023.

[99] 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 (ICCV)*, pages 5826–5835, October 2021.

[100] Haoyi Zhu, Honghui Yang, Xiaoyang Wu, Di Huang, Sha Zhang, Xianglong He, Tong He, Hengshuang Zhao, Chunhua Shen, Yu Qiao, et al. Ponderv2: Pave the way for 3d foundataion model with a universal pre-training paradigm. *arXiv preprint arXiv:2310.08586*, 2023.

[101] Joseph Zhu and Peiye Zhuang. Hifa: High-fidelity text-to-3d with advanced diffusion guidance. *arXiv preprint arXiv:2305.18766*, 2023.

[102] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2639–2650, 2023.

A Appendix

The appendix is organized as follows. First, we introduce the data source of our method (Appendix A.1), and the data organization and processing pipeline in Appendix A.2. After that, we show that the mesh generated by MeshXL is compatible with existing texturing methods for high-quality 3D assets in Appendix A.4. We also put forward discussions in Appendix A.5. For more visualization results, please see our attachment for videos.

A.1 Data Sources

We provide additional details on the 3D data sources we use to train and evaluate our models.

ShapeNet V2 [9] collects about 51k 3D CAD models for 55 categories. We split the data in 9:1 for training and validation by each category.

3D-FUTURE [22] present about 10k high-quality 3D mesh data for indoor furniture. However, because of the delicate design, the objects contain many faces. Therefore, only a small proportion of the data can be used to train our MeshXL models.

Objaverse [17] is a large 3D data collection with more than 800k 3D objects for about 21k categories collected from Sketchfab. We split the data in 99:1 for training and validation, respectively.

Objaverse-XL [16] further expand Objaverse [17] into a dataset with more than 10M 3D objects with additional data collected from GitHub, Polycam, Thingiverse, and Smithsonian. We split the Github and Thingiverse part of the Objaverse-XL dataset into 99:1 for training and validation, respectively.

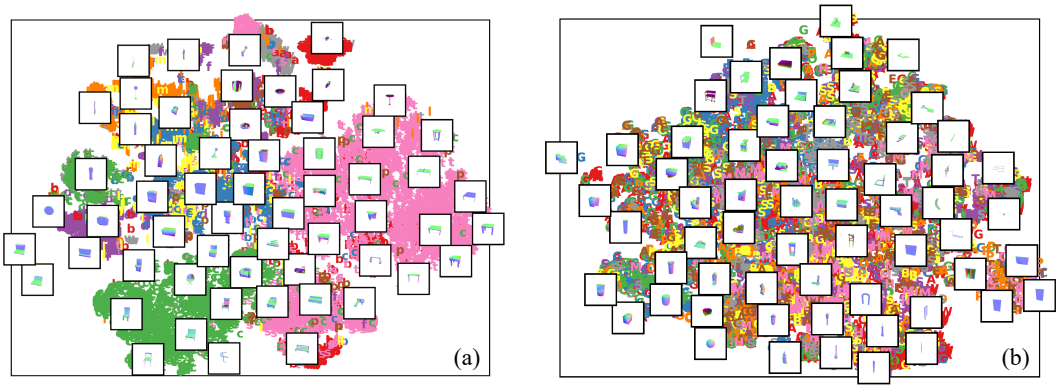


Figure 8: We visualize the text-annotated 3D dataset using t-SNE to illustrate the primary structure of the training data. (a) The ShapeNet dataset, the general 3D baseline dataset. (b) The filtered Objaverse-XL dataset, the large open-source 3D dataset.

A.2 Data

Data collection and filtering. To organize existing datasets, we build up a filtering and pre-processing pipeline to ensure that the meshes met our demand. We first collect meshes with fewer than 800 faces, and ensure that they have corresponding UV maps for rendering. After that, we render the 3D meshes, and discard those are not center-aligned or occupying less than 10% of the frame. For those 3D meshes with more than 800 but less than 20,000 faces, we use planar decimation whether their meshes can be simplified. Finally, we achieve approximately 2.5 million pieces of data remained.

Planar Decimation Pipeline. To ensure the quality of the decimated 3D meshes, we make sure either a lower Hausdorff distance $\delta_{\text{hausdorff}}$ [66] or a similar rendered views [11].

Collecting mesh-text pairs. We first render each 3D mesh with 12 different views, and concatenate them into one single image. Then, we annotate both the front view image and the fused multi-view image using CogVLM [77]. After that, we adopt the Mistral-7B-Instruct model [35] with few-shot in-context examples to extract information on category and geometry from the CogVLM annotations. We tag each 3D mesh with the resulting categories and 3 to 5 geometry descriptors.

Collecting mesh-image pairs. To produce diverse image conditions for 3D mesh generation, we first generate images with multi-view image and depth rendering. After that, we use the sentences produced by CogVLM [77] as the prompt, and use a find-tuned Stable Diffusion model [63] to augment the rendered images for diverse textures and backgrounds. To ensure the quality of the generated images, we also adopt a manually cleansing procedure.

Data Statistics. We present data statistics of our collected and organized data in Tab. 5.

Table 5: **Statistics for the Training Data and Validation Data.** After combining four data sources, our proposed MeshXL models are trained on approximately 2.5 million 3D meshes.

Dataset	Pre-training		Text-to-3D	
	Train	Val	Train	Val
ShapeNet [9]	16,001	1,754	15,384	1,728
3D-Future [22]	1,603	-	-	-
Objaverse [17]	85,282	854	83,501	820
Objaverse-XL [16]	2,407,337	15,200	1,347,802	13,579
Total	2,510,223	17,808	1,446,678	16,127

A.3 Image/Text to Mesh

We provide details on how we enable MeshXL models to generation high-fidelity 3D assets given the additional image or text as the condition.

Condition Encoder. We adopt a pre-trained BERT [18] model for text condition encoding, and a pre-trained ViT [19] model for image feature encoding.

Condition Injection. To align the additional text/image feature with the mesh coordinate field, we adopt the Q-Former architecture [40] to compress the encoded feature into a fixed-length of 32 learnable tokens as the prefix of the MeshXL model. The model is trained to generate mesh tokens given the condition prefix (see Eq. (2)).

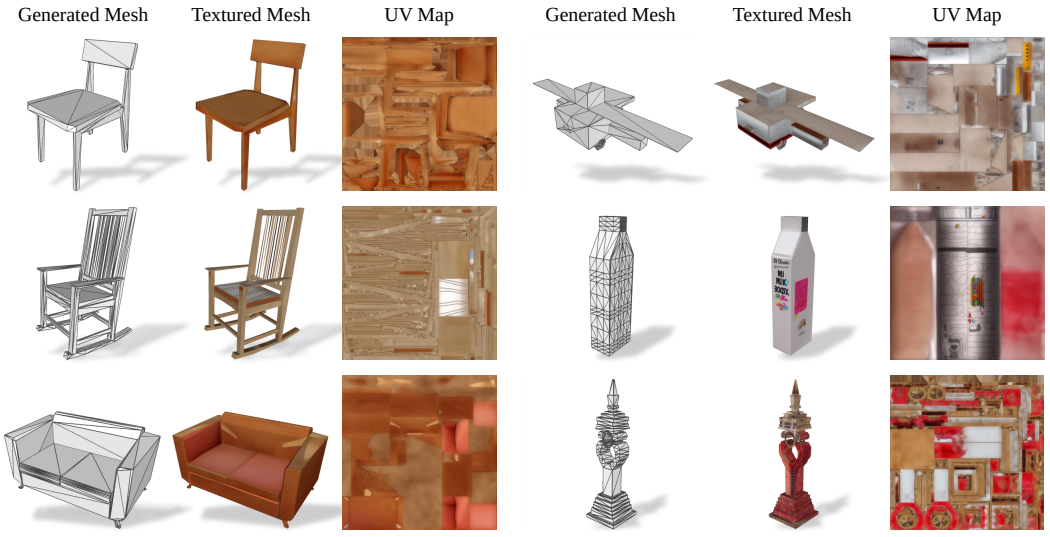


Figure 9: **Texture Generation for the Generated 3D Meshes.** We adopt Paint3D [93] to generate textures for 3D meshes produced by MeshXL.

A.4 Texturing

We show the generated meshes by MeshXL and generate textures in Fig. 9. Specifically, faces are unwrapped to a UV map using Xatlas, then texture maps are generated by the coarse-to-fine

generation pipeline introduced by Paint3D [93]. We found that finetuning the UNet decoder of the controlled diffusion model can enhance the texture map generation stability.

A.5 Discussions

Difference with PolyGen [53]. PolyGen explores the auto-regressive generation of 3D polynomial meshes with two transformers [74], *i.e.* the *vertex transformer* and the *face transformer*. PolyGen first generates a set of points representing the vertices of the 3D meshes with a vertex transformer. After that, PolyGen inputs the generated point cloud into the face transformer and predicts the connectivity among the generated with a face transformer. However, our proposed MeshXL is a more straightforward and end-to-end approach that directly generates the polynomial meshes auto-regressively with decoder-only transformers.

Difference with MeshGPT [66]. MeshGPT consists of a mesh VQVAE [73] and a decoder-only transformer [59]. MeshGPT first learns a mesh VQVAE to quantize the 3D meshes into discrete tokens. After that, MeshGPT trains a decoder-only transformer to generate the discrete tokens for 3D mesh reconstruction. In comparison, our proposed MeshXL is an end-to-end method that learns the neural representation of coordinates and outputs 3D meshes directly.

Extensibility. Our method, MeshXL, is built upon the concept of auto-regressive methods. Therefore, our method is not restricted to the decoder-only transformers [59, 95, 70, 71], and can also be extended to other causal language models (*i.e.* Mamba [26], RWKV [55], and xLSTM [6]).