SEEDFACE: RETHINKING AUTOREGRESSIVE MESH INITIALIZATION THROUGH FIRST-FACE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive models have shown strong potential for native 3D mesh generation, but their performance degrades on broad and diverse shape distributions. We show that this difficulty is highly non-uniform across the generation process. In particular, autoregressive mesh generation suffers from an initialization bottleneck: the first face must establish geometric context from an empty history, while later faces are strongly constrained by previously generated geometry. Prefix experiments show that providing only the first ground-truth face substantially improves generation, and further analysis reveals that the first face has the highest prediction uncertainty and becomes increasingly difficult to model as shape diversity grows. Motivated by this asymmetry, we propose *SeedFACE*, which separates geometric initialization from autoregressive continuation. We factor mesh generation into a seed distribution and a conditional continuation model. Rather than generating the seed face token by token from an empty context, SeedFACE represents it in a regularized latent space and learns its distribution with a lightweight latent generative model. The sampled seed is then decoded by the shared mesh decoder and serves as the geometric context for standard autoregressive generation. This preserves autoregression where geometric context is informative while avoiding token-by-token prediction in the least constrained regime. Experimental results on Objaverse demonstrate that SeedFACE substantially improves generation quality and diversity over the state-of-the-art autoregressive baselines.

1 INTRODUCTION

Recent autoregressive models have made notable progress in native 3D mesh generation, directly modeling meshes as sequences of geometric tokens and producing high-quality surfaces without relying on instance-specific 3D guidance (Nash et al., 2020; Siddiqui et al., 2024; Chen et al., 2024; Weng et al., 2024; Lei et al., 2026; Wang et al., 2026a). While these models perform well on category-specific datasets such as ShapeNet (Chang et al., 2015), their performance degrades when extended to broader and more diverse shape distributions such as Objaverse (Deitke et al., 2023), particularly in terms of generation quality and diversity. This raises a fundamental question: *is this degradation an inherent limitation of autoregressive mesh modeling, or is the difficulty concentrated at a particular stage of the causal generation process?*

We investigate this question with a simple prefix intervention. Given a trained autoregressive mesh generator, we provide a small number of ground-truth prefix faces from the test set and let the model generate the remaining mesh autoregressively. Surprisingly, as shown in Fig. 1, providing only the first face already yields substantial improvements for both PivotMesh (Weng et al., 2024) and FACE (Wang et al., 2026a) under the category-agnostic nature of Objaverse, whereas providing additional prefix faces brings comparatively limited gains. The qualitative evidence also supports this finding, where geometric fragments that are rarely reached when generation starts from scratch become accessible once a single initial face is supplied. This result reveals an important asymmetry in autoregressive mesh generation. The model is substantially more capable of *continuing* a mesh once a geometric context has been established than of *initializing* that context from an empty history.

Our analysis further traces this asymmetry to a clear *first-face modeling bottleneck*. The first face exhibits substantially higher prediction uncertainty than subsequent faces because it must establish geometry without any preceding spatial context and consists entirely of newly introduced vertices.

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.

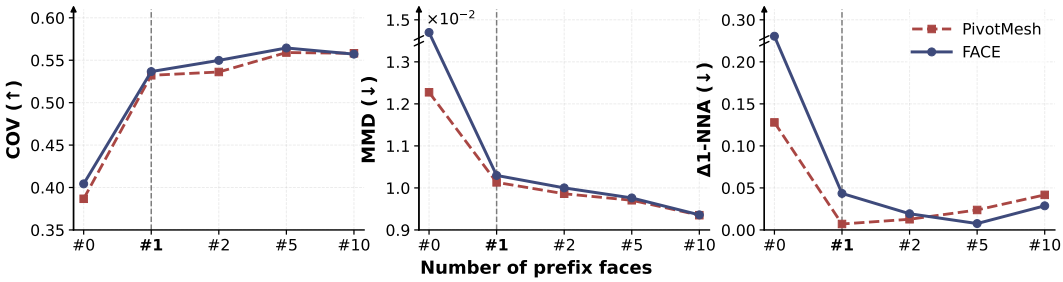


Figure 1: **Investigating the Prefix-face effects of PivotMesh and FACE on Objaverse.** In addition to COV and MMD, We report Δ1-NNA $=$ 1-NNA-50%, where values closer to zero indicate better performance. Conditioning on only the first ground-truth face (#1) substantially improves all metrics over unconditional generation (#0), while additional prefix faces yield smaller gains. These results highlight the importance of first-face initialization in autoregressive mesh generation.

Once this initial geometry is available, the mesh traversal becomes increasingly constrained. Later faces frequently reuse existing vertices, and most remaining uncertainty is associated only with newly introduced geometry. More importantly, this initialization problem becomes harder as the shape distribution grows more diverse. On Objaverse, the first-face distribution spans a substantially broader set of geometric configurations, and the autoregressive model exhibits a larger modeling gap beyond the data's intrinsic uncertainty. Thus, scaling to diverse shape distributions does not merely make every autoregressive prediction uniformly harder. It disproportionately increases the difficulty of choosing a plausible geometric starting point from which the rest of the mesh can be generated.

This observation motivates a different treatment of initialization and continuation. We refer to the first face that establishes the initial geometric context as the *seed face*, and propose **SeedFACE**, which explicitly models its distribution while retaining autoregressive generation for the remainder of the mesh. Conceptually, SeedFACE factorizes mesh generation into two regimes: an underconstrained initialization problem, in which a coherent geometric seed must be selected without prior context, and a constrained continuation problem, in which subsequent geometry is generated conditioned on an increasingly informative partial mesh. Rather than forcing the causal predictor to construct the seed token by token from an empty history, we encode the seed face into a regularized latent space (Kingma & Welling, 2013) and learn its distribution with a lightweight latent generative model. A diffusion model (Ho et al., 2020; Rombach et al., 2022) is then used to sample from this latent distribution. The sampled latent is decoded into a coherent seed through the shared mesh-generation pathway, after which the original autoregressive process continues from the generated seed face. In this way, SeedFACE changes the modeling mechanism precisely where autoregression is least constrained, while preserving the causal structure that is already effective for subsequent mesh generation.

Experiments on large-scale mesh generation support this initialization-centered view. On Objaverse, SeedFACE improves 1-NNA from 76.10 to 53.56, COV from 40.43 to 52.88, and reduces MMD from 14.40 to 10.46 compared with the autoregressive baseline. Performance approaches that of an oracle supplied with a ground-truth first face, suggesting that learned initialization can recover much of the benefit of a valid geometric starting point. Qualitative comparisons further suggest broader geometric coverage and fewer generation failures due to poor initial configurations. Together, these results suggest that improving the scalability of autoregressive mesh generation does not necessarily require replacing autoregression throughout the entire sequence. Instead, a key opportunity lies in better modeling the geometric state from which causal generation begins.

In summary, our contributions are threefold:

- We identify an *initialization bottleneck* in autoregressive mesh generation, showing that the first face is substantially harder to model than subsequent faces.
- We show that this bottleneck becomes more severe as shape diversity increases, with broader first-face distributions and larger autoregressive modeling gaps.
- We propose *SeedFACE*, which models the seed-face distribution in a regularized latent space while preserving autoregressive generation of subsequent faces, resulting in higher quality and diversity.

2 Related Work

3D Generative Modeling. Early 3D generative models learned shape distributions using volumetric, point-based, and implicit representations (Wu et al., 2015; 2016; Yang et al., 2019; Chen & Zhang, 2019). With the emergence of neural radiance fields and Gaussian splatting (Mildenhall et al., 2021; Kerbl et al., 2023), pretrained image diffusion models were increasingly leveraged for 3D generation through score distillation, multi-view guidance, explicit-surface refinement, and Gaussian-based generation (Poole et al., 2022; Wang et al., 2023a; Metzer et al., 2023; Lin et al., 2023; Chen et al., 2023; Wang et al., 2023b; Shi et al., 2024; Zhu et al., 2024; Qiu et al., 2024; Tang et al., 2024; Yi et al., 2024; Liang et al., 2024). More recently, large-scale 3D-native models have learned aligned, structured, or spatially sparse latent representations with diffusion or flow-based generative models, substantially improving scalability, controllability, and geometric fidelity (Zhao et al., 2023; Zhang et al., 2024; Xiang et al., 2025b; Zhao et al., 2025b; Li et al., 2025b;a; Wu et al., 2026; Lai et al., 2026; Xiang et al., 2025a). Despite these advances, polygonal meshes are commonly obtained through a subsequent decoding, extraction, or conversion stage rather than being modeled directly as structured mesh elements.

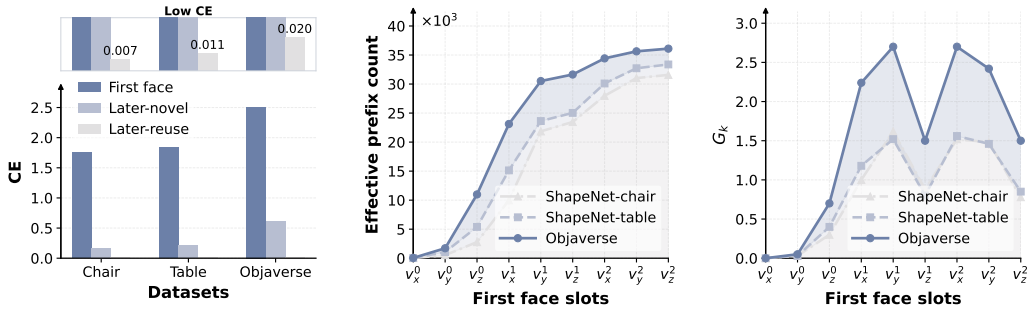
Geometry-Guided Mesh Generation. A growing line of work generates structured meshes with guidance from existing or intermediate 3D geometry, such as point clouds, surfaces, signed distance fields, or sparse geometric scaffolds. Autoregressive approaches and related hybrid designs improve mesh tokenization, locality, generation order, and scalability while conditioning the generation process on geometric observations (Chen et al., 2025a;b; Weng et al., 2025; Lionar et al., 2025; Liu et al., 2025; Wang et al., 2025; Zhao et al., 2025a; Tang et al., 2025; Kim et al., 2026; Hao et al., 2024; Lin et al., 2026; Wang et al., 2026a; Xu et al., 2026b;c). Beyond fully serialized autoregressive decoding, other approaches employ continuous connectivity representations, part-wise diffusion, latent flow matching, or topology-aware geometric fields under similar geometric guidance (Shen et al., 2024; Yang et al., 2026; Li et al., 2026b;a; Zhao et al., 2026; Long et al., 2026; Wang et al., 2026b; Xu et al., 2026a). Together, these efforts enable structured mesh generation from existing 3D information, providing an effective route for remeshing and post-processing proxy 3D representations.

Native Mesh Generation. Native mesh generation instead models polygonal mesh distributions without instance-specific 3D guidance. Autoregressive methods range from vertex- and face-sequence modeling to learned mesh tokens, neural coordinate representations, hierarchical generation, and progressive levels of detail (Nash et al., 2020; Siddiqui et al., 2024; Chen et al., 2024; Weng et al., 2024; Lei et al., 2026). In parallel, non-autoregressive approaches directly model mesh geometry and topology through discrete diffusion or continuous flow-based generation (Alliegro et al., 2023; He et al., 2025; Sun et al., 2026). Our work focuses on autoregressive native mesh generation, where generation proceeds sequentially from previously generated geometry. Existing methods largely study mesh representations and generation orders, while the causal structure of autoregressive generation, *particularly how the initial prediction influences subsequent decoding*, remains underexplored.

3 Analysis of First-Face Modeling Difficulty

Providing a single ground-truth face substantially improves generation, with limited gains from additional prefix faces (Fig. 1). We investigate this asymmetry through three aspects: first-face prediction uncertainty, distribution diversity, and modeling gap.

Prediction Uncertainty of First-Face. To explore how prediction difficulty varies with geometric context in autoregressive mesh generation, we compare per-vertex prediction uncertainty between the first and subsequent faces via cross-entropy. The vertices of the latter faces are further separated into *novel vertices* that introduce new geometry and *reused vertices* from the already-generated mesh. Fig. 2a shows that the first face has substantially higher prediction uncertainty. For subsequent faces, novel vertices account for most of the remaining prediction uncertainty, while reused vertices have very low cross-entropy. The low score of reused vertices suggests that the preceding sequence strongly constrains their prediction, while the higher cross-entropy of novel vertices highlights the difficulty of introducing new geometry. This difficulty is particularly pronounced for the first face, where all vertices are generated without any preceding spatial or geometric context.



(a) Cross-entropy comparison (b) First-face prefix diversity (c) First-face modeling gap

Figure 2: **The First-Face Modeling Bottleneck.** (a) Cross-entropy (CE) for the first face, novel vertices in later faces, and reused vertices in later faces. The first face is substantially harder to model, especially on Objaverse, whereas reused vertices exhibit low CE (the upper plot). (b) First-face prefix diversity, measured by the effective prefix count across the nine token positions. Objaverse exhibits substantially greater prefix diversity than ShapeNet *chair* and *table*. (c) First-face modeling gap, measured by G_k. Objaverse exhibits a larger modeling gap across the first-face sequence, indicating increased difficulty in capturing its more diverse first-face distribution.

Distribution Diversity of First-Face. We next examine how the first-face distribution varies with increasing data scale and diversity. We represent each first face as nine consecutive coordinate tokens and measure the effective number of observed prefixes $N_{\text{eff}}(k)$, at each token position k. As shown in Fig. 2b, Objaverse exhibits substantially greater prefix diversity than the *chair* and *table* subsets of ShapeNet (Chang et al., 2015), with the effective prefix space expanding more rapidly as additional coordinates are specified. Therefore, scaling to Objaverse requires the autoregressive model to establish its initial context from a substantially broader and less concentrated distribution of geometries, making initialization increasingly challenging as shape diversity grows.

Modeling Gap of First-Face. A higher prediction cross-entropy can reflect both greater intrinsic uncertainty in the first-face distribution and a larger discrepancy between the model and that distribution. To distinguish these contributions, we compare the teacher-forced model cross-entropy CE_k with the empirical conditional entropy computed from dataset token frequencies at each position k:

$$G_k = CE_k - \mathcal{H}(X_k \mid X_{<k}), \tag{1}$$

where both quantities condition on the ground-truth prefix $X_{<k}$. The resulting gap measures excess cross-entropy beyond the uncertainty estimated from the data. As shown in Fig. 2c, the gap is substantially larger on Objaverse than on the ShapeNet subsets, suggesting that the model struggles to capture the first-face distribution beyond the difficulty attributable to its intrinsic uncertainty.

In summary, these findings identify first-face initialization as a bottleneck that becomes more pronounced with increasing shape diversity, motivating a dedicated model for the seed-face distribution. Further analyses of prediction uncertainty, distribution complexity, modeling gaps, and empirical coverage are provided in Appendix A.

4 SeedFACE

Motivated by the first-face modeling difficulty identified above, we propose *SeedFACE*, which models the first-face distribution in a regularized latent space and uses the sampled face to initialize autoregressive continuation. The overall pipeline is illustrated in Fig. 3.

4.1 Seed-Conditioned Autoregressive Modeling

Let $\mathcal{F} = (F_1, \ldots, F_T)$ denote the complete face sequence. We refer to the first T_s faces, $\mathcal{S} = \mathcal{F}_{1:T_s} = (F_1, \ldots, F_{T_s})$, as the *seed faces*, which provide the geometric initialization for subsequent generation. Following the analysis in Sec. 3, we use $T_s = 1$ by default. The extension to $T_s > 1$ is

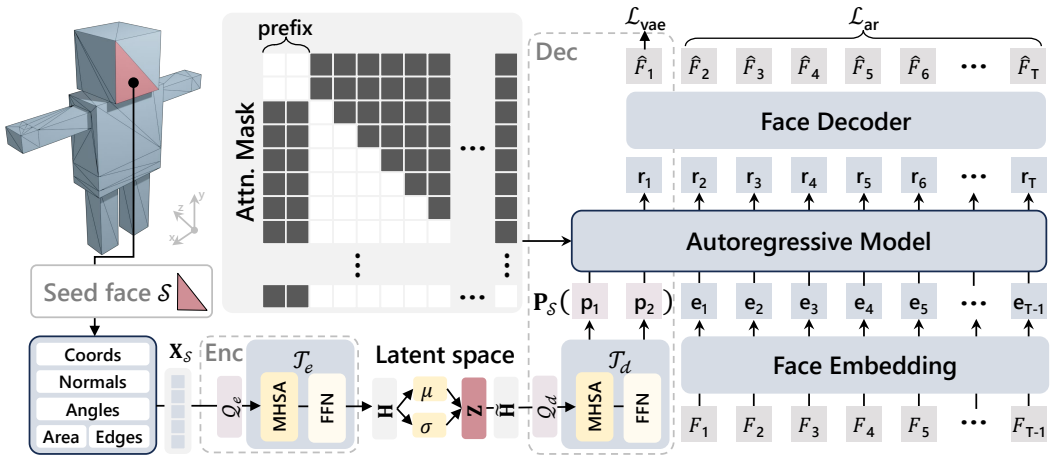


Figure 3: **Overview of SeedFACE.** We model the seed-face distribution in a regularized latent space using a VAE. The decoded latent prefix is passed through the shared autoregressive model and face decoder to reconstruct the seed face, while subsequent faces are generated with the original causal structure conditioned only on previously generated faces. The figure illustrates the case of $T_s = 1$.

described in Appendix B.1. Once plausible seed faces are established, the original causal model can effectively continue the remaining sequence. We therefore factorize the mesh distribution as

$$p(\mathcal{F}) = p_{\text{seed}}(\mathcal{S}) \prod_{t=T_s+1}^{T} p_\theta(F_t \mid F_{<t}), \tag{2}$$

where $p_{\text{seed}}(\mathcal{S})$ denotes the seed-face distribution modeled by a latent generative model, while p_θ denotes the original autoregressive predictor retained for subsequent faces.

SeedFace VAE. To obtain a compact representation of the seed faces $\mathcal{S} = \mathcal{F}_{1:T_s}$, we introduce a lightweight VAE (Kingma & Welling, 2013) that is directly integrated with the original face-based autoregressive model (Wang et al., 2026a). We first construct the seed-face token sequence $\mathbf{X}_\mathcal{S} = \text{Tok}(\mathcal{S})$ from geometric attributes including quantized coordinates, angles, edges, normals, and face areas. To extract global context, N_e learnable encoder queries $\mathbf{Q}_e$ are concatenated with the seed-face tokens and jointly processed by a stack of Transformer blocks. Only the output features corresponding to the encoder queries are retained:

$$\mathbf{H} = \text{Enc}(\mathbf{X}_\mathcal{S}) = [\mathcal{T}_e([\mathbf{Q}_e; \mathbf{X}_\mathcal{S}])]_{\mathbf{Q}_e}, \qquad \mathbf{H} \in \mathbb{R}^{N_e \times D}, \tag{3}$$

where $\mathcal{T}_e$ denotes the first-face encoder Transformer and $[\cdot]_{\mathbf{Q}_e}$ selects the output positions corresponding to the encoder queries.

The encoded features $\mathbf{H}$ are then linearly projected to a lower-dimensional C-channel space, from which the mean $\boldsymbol{\mu}$ and log-variance $\log \boldsymbol{\sigma}^2$ of the latent distribution are predicted. The latent representation is sampled using the reparameterization trick:

$$\mathbf{Z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\eta}, \qquad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \mathbf{Z} \in \mathbb{R}^{N_e \times C}. \tag{4}$$

The latent distribution is regularized toward a standard Gaussian using KL divergence. For decoding, $\mathbf{Z}$ is linearly projected back to the Transformer dimension, producing $\widetilde{\mathbf{H}} \in \mathbb{R}^{N_e \times D}$.

We then concatenate $\widetilde{\mathbf{H}}$ with N_d learnable decoder queries $\mathbf{Q}_d$ and jointly process them with the first-face decoder Transformer $\mathcal{T}_d$. The outputs corresponding to the decoder queries are retained as object-specific prefix tokens:

$$\mathbf{P}_\mathcal{S} = \left[\mathcal{T}_d\left([\widetilde{\mathbf{H}}; \mathbf{Q}_d]\right)\right]_{\mathbf{Q}_d}. \tag{5}$$

These prefix tokens are fed into the original autoregressive predictor to generate face features, which are subsequently decoded by the causal face decoder. We collectively denote the entire decoding

process as

$$\widehat{\mathcal{S}} = \mathrm{Dec}(\widetilde{\mathbf{H}}) = \mathrm{FaceDec}\left(\mathrm{AR}\left(\mathbf{P}_{\mathcal{S}}\right)\right). \tag{6}$$

In this way, the modeling of the seed-face distribution is shifted from direct autoregressive prediction to a stochastic latent space $\mathbf{Z}$. Instead of requiring the autoregressive model to directly capture the highly diverse first-face distribution, the VAE learns a compact latent representation in which this distribution can be modeled more effectively. The sampled latent is then mapped to a prefix sequence and decoded through the original autoregressive predictor AR and causal face decoder FaceDec. These modules therefore primarily serve as a shared geometric decoder that realizes the latent-conditioned seed faces, rather than modeling their stochastic distribution directly. By introducing this latent space while largely preserving the original decoding pathway, the VAE provides a more expressive and structured mechanism for modeling the first face.

Shared Decoding with Preserved Causality. The shared AR–FaceDec pathway is used differently for seed-face reconstruction and subsequent autoregressive generation. For the seed faces, the latent representation is converted into a prefix sequence $\mathbf{P}_{\mathcal{S}} = (\mathbf{p}_1, \ldots, \mathbf{p}_{N_d})$. All prefix tokens are mutually visible, and the last T_s prefix positions serve as prediction slots for the T_s seed faces. The shared autoregressive predictor therefore produces their representations in parallel:

$$[\mathbf{r}_1, \ldots, \mathbf{r}_{T_s}] = \left[\mathrm{AR}\left(\mathbf{P}_{\mathcal{S}}\right)\right]_{N_d - T_s + 1 : N_d}, \qquad \widehat{F}_t = \mathrm{FaceDec}(\mathbf{r}_t), \quad 1 \le t \le T_s. \tag{7}$$

Thus, the VAE reconstructs the seed faces through the same AR and FaceDec modules as the original generator, but uses a prefix-visible attention pattern that allows all seed faces to be decoded in parallel. Following FACE (Wang et al., 2026a), each generated face F_t is mapped by the face embedding module to a face embedding $\mathbf{e}_t$, which is appended to the autoregressive context for subsequent generation. The remaining faces are then generated causally from the embeddings of previously generated faces using the original AR–FaceDec pipeline. Therefore, the latent representation influences the subsequent mesh sequence only through the generated seed faces, while subsequent generation follows the original causal decoding process of FACE. The attention map is illustrated in Fig. 3.

Training Objective. We jointly optimize the SeedFace VAE and the autoregressive mesh model. The reconstruction loss is computed as token-level cross-entropy between each predicted face $\widehat{F}_t$ and its ground-truth counterpart F_t under the underlying discrete face representation. The VAE objective consists of the reconstruction loss over the T_s seed faces and the KL regularization of the latent space:

$$\mathcal{L}_{\mathrm{vae}} = \sum_{t=1}^{T_s} \mathrm{CE}\left(\widehat{F}_t, F_t\right) + \beta_{\mathrm{kl}} D_{\mathrm{KL}}\left[q(\mathbf{Z} \mid \mathcal{S}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\right]. \tag{8}$$

For the remaining faces, we retain the original autoregressive cross-entropy objective:

$$\mathcal{L}_{\mathrm{ar}} = \sum_{t=T_s+1}^{T} \mathrm{CE}\left(\widehat{F}_t, F_t\right). \tag{9}$$

The overall training objective is

$$\mathcal{L}_{\mathrm{seed}} = \mathcal{L}_{\mathrm{ar}} + \lambda_{\mathrm{vae}} \mathcal{L}_{\mathrm{vae}}, \tag{10}$$

where λ_{vae} balances the seed-face VAE objective with the original autoregressive objective.

4.2 SeedFace Latent Diffusion

After training the SeedFace VAE, we model the distribution of its latent representations $\mathbf{Z} \in \mathbb{R}^{N_e \times C}$ using a lightweight latent diffusion model (Ho et al., 2020; Rombach et al., 2022). Given a clean latent $\mathbf{Z}_0$, the forward diffusion process adds Gaussian noise at timestep τ:

$$\mathbf{Z}_\tau = \sqrt{\bar{\alpha}_\tau}\mathbf{Z}_0 + \sqrt{1 - \bar{\alpha}_\tau}\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{11}$$

where $\bar{\alpha}_\tau$ is determined by the predefined noise schedule. A denoising network ϵ_ψ is trained to predict the injected noise:

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{\mathbf{Z}_0, \tau, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \epsilon_\psi(\mathbf{Z}_\tau, \tau)\right\|_2^2\right]. \tag{12}$$

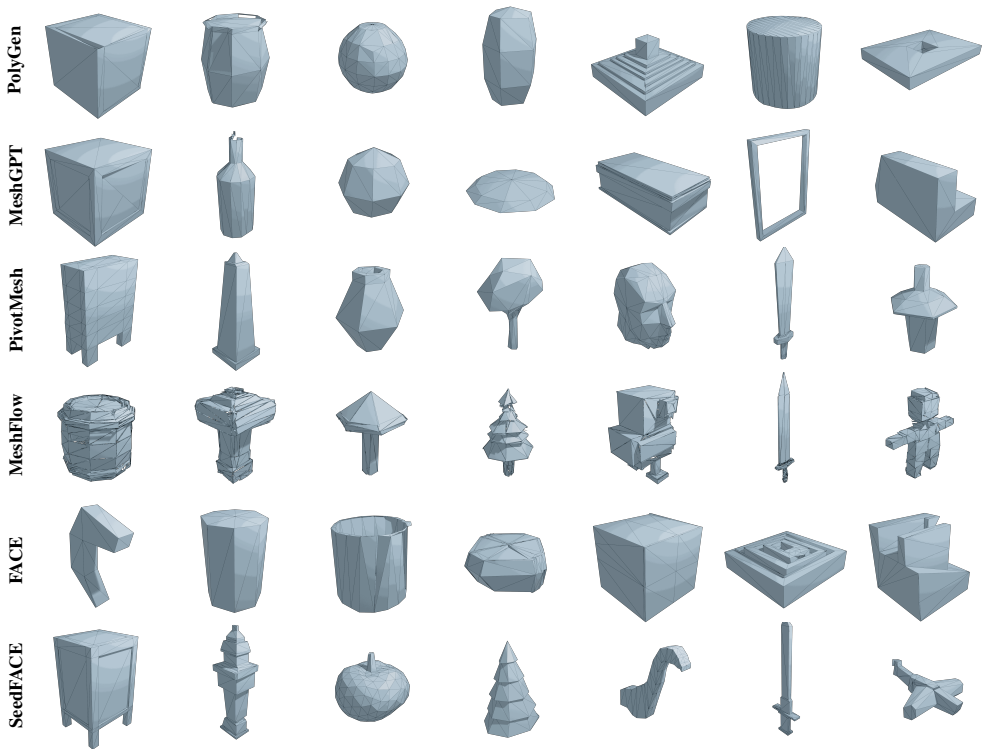


Figure 4: Qualitative comparison of category-agnostic mesh generation on Objaverse.

Table 1: Quantitative comparison on Objaverse. COV, MMD, and 1-NNA are reported in $\%$, 10^{-3}, and $\%$, respectively. For 1-NNA, values closer to 50% are better. The best and second-best results are highlighted in **bold** and underline, respectively.

Metric	MeshFlow	PolyGen	MeshGPT	PivotMesh	FACE	SeedFACE
COV↑	<u>51.86</u>	40.66	26.41	38.67	40.43	**52.88**
MMD↓	12.33	<u>11.38</u>	15.80	12.28	14.40	**10.46**
1-NNA	<u>54.54</u>	61.11	82.16	62.79	76.10	**53.56**

At inference, we sample a latent from Gaussian noise through the reverse diffusion process, decode it into the seed faces, and continue the remaining mesh generation using the original causal autoregressive model.

4.3 Extensions of Seed Representation

Extension to Other Autoregressive Baseline. We further adapt SeedFACE to PivotMesh (Weng et al., 2024), whose first face is represented by six discrete tokens. The same seed modeling formulation is applied to these tokens, followed by the original autoregressive generation process.

Lightweight Plugin. For a pretrained autoregressive model, the seed generator can also be used as a lightweight plugin without retraining the full architecture. Instead of the latent VAE, the initial seed can be modeled directly in the original data space. We implement continuous diffusion and flow matching (Lipman et al., 2022; Liu et al., 2023) for seed geometry, while discrete models such as D3PM (Austin et al., 2021) can similarly operate on quantized seed tokens. The generated seed is then fed to the pretrained AR model to complete the remaining sequence. Detailed implementations are provided in Appendix B.

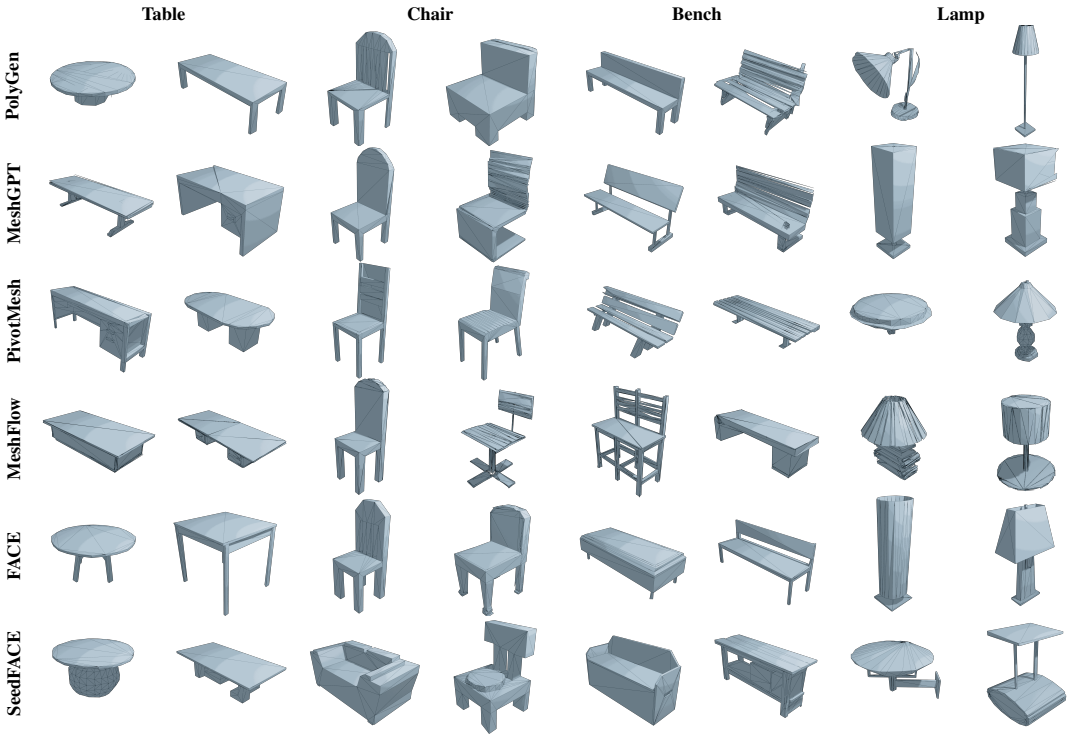


Figure 5: Qualitative comparison of category-conditioned mesh generation on ShapeNet. Each pair of columns presents generated meshes for the same object category.

Table 2: Quantitative comparison on ShapeNet. COV, MMD, and 1-NNA are reported in %, 10^{-3}, and %, respectively. For 1-NNA, values closer to 50% are better. Results shown in gray are taken from MeshGPT (Siddiqui et al., 2024) and included for reference only. The best and second-best results from MeshFlow to SeedFACE are highlighted in **bold** and underline, respectively.

Method	Chair			Table			Bench			Lamp		
	COV↑	MMD↓	1-NNA	COV↑	MMD↓	1-NNA	COV↑	MMD↓	1-NNA	COV↑	MMD↓	1-NNA
AtlasNet	9.03	4.05	95.13	7.16	3.85	96.30	20.53	2.47	90.58	19.97	4.68	91.85
BSPNet	16.48	3.62	91.75	16.83	3.14	93.58	28.74	2.05	88.44	18.38	5.32	93.13
GET3D	40.85	3.56	83.04	41.70	2.78	85.54	–	–	–	–	–	–
MeshFlow	51.92	9.59	53.46	54.07	9.05	54.18	51.85	10.01	58.33	60.00	21.62	**51.25**
PolyGen	50.38	11.32	65.77	51.15	10.60	65.24	**55.56**	9.69	54.63	52.50	24.30	53.75
MeshGPT	33.85	11.95	71.35	34.24	14.58	76.41	38.89	12.18	76.85	30.00	35.69	76.25
PivotMesh	40.38	11.06	64.43	47.18	10.71	61.90	50.00	11.93	53.70	**67.50**	22.97	52.50
FACE	37.69	11.52	64.42	43.01	11.01	57.52	46.30	10.03	59.26	45.00	**21.54**	52.50
SeedFACE	**54.23**	**9.50**	**50.19**	**55.11**	**9.04**	**52.61**	46.30	**9.51**	**50.00**	55.00	22.58	52.50

5 Experiments

5.1 Experiment Settings

Datasets. We evaluate on ShapeNetV2 (Chang et al., 2015) and Objaverse (Deitke et al., 2023). Following prior work (Siddiqui et al., 2024; Weng et al., 2024), we use the chair, table, bench, and lamp categories of ShapeNet. After planar decimation, we retain meshes with fewer than 800 faces for ShapeNet and 500 faces for Objaverse, which also serve as the maximum sequence lengths for autoregressive training. Both datasets are split into training and test sets with a 9:1 ratio.

Metrics. We measure distribution-level generation quality with Coverage (COV), Minimum Matching Distance (MMD), and 1-Nearest-Neighbor Accuracy (1-NNA), following (Siddiqui et al., 2024; Weng et al., 2024). COV reflects the diversity of generated samples, MMD measures their prox-

imity to the reference distribution, and 1-NNA assesses the separability between generated and real samples.

Implementation Details. Following (Siddiqui et al., 2024; Weng et al., 2024), we augment meshes with random scaling and translation and quantize vertex coordinates to 7 bits per axis. For the seed-face VAE, both the encoder and decoder Transformers consist of 2 layers. We set the numbers of encoder and decoder queries to $N_e = N_d = 2$, respectively, with each latent token having a dimension of 32, resulting in a latent representation of size 2×32. The autoregressive backbone follows the architecture of PivotMesh (Weng et al., 2024), using a 24-layer Transformer with a hidden dimension of 1024. Following FACE (Wang et al., 2026a), the face decoder is implemented as a 2-layer causal MLP. The latent diffusion model is implemented using a 16-layer Transformer.

5.2 Quantitative Evaluation

Quantitative results on Objaverse and ShapeNet are reported in Table 1 and Table 2, respectively.

On Objaverse, SeedFACE achieves clear improvements over the autoregressive baselines, increasing COV from approximately 40% to 52.88% while bringing 1-NNA closer to the ideal value of 50%. These results indicate improved generation diversity and a reduced distributional gap between generated and reference meshes. SeedFACE also outperforms the flow-based baseline MeshFlow under the adopted evaluation metrics. We note, however, that these metrics are computed on point clouds sampled from the generated meshes and therefore do not fully capture geometric discontinuities or connectivity artifacts. Such issues are more clearly revealed by the qualitative comparisons in Fig. 4.

On ShapeNet, SeedFACE achieves competitive performance across all four categories. For chair and table, which contain relatively larger numbers of training samples, SeedFACE generally provides an advantage over the compared baselines. For bench and lamp, the differences among methods are smaller because the filtered subsets are relatively small and can be well fitted by all compared methods. Overall, the results suggest that the advantage of SeedFACE becomes more evident as the scale and diversity of the training distribution increase.

5.3 Qualitative Evaluation

Qualitative comparisons on Objaverse and ShapeNet are shown in Fig. 4 and Fig. 5, respectively. Across both datasets, SeedFACE produces a broader range of geometries and is better able to capture less frequent shape structures. In comparison, the autoregressive baselines more frequently generate simple box-like or cube-like geometries, whereas SeedFACE produces more diverse object structures and geometric configurations. These observations are particularly evident on Objaverse, where the underlying shape distribution is more diverse. Additional comparisons and generation galleries are provided in Appendix C.4.

5.4 Additional Experiments

We further evaluate lightweight seed-modeling plugins, the adaptation of SeedFACE to PivotMesh, different numbers of seed faces, and the VAE encoder ablation in Appendix C.

6 Conclusion

In this paper, we present an initialization-centered perspective on autoregressive mesh generation, identifying the establishment of geometric context as a critical bottleneck. This perspective emerges from the finding that predicting the first face from an empty history is substantially harder than continuing an existing mesh, with the disparity growing as shape diversity increases. Guided by this insight, we propose SeedFACE, which factorizes mesh generation into a seed distribution and a conditional continuation model. A regularized latent representation recasts seed-face construction as latent distribution modeling, enabling coherent initialization followed by standard autoregressive generation. Experiments demonstrate improved generation quality and diversity, supporting the effectiveness of the first face initialization. These findings suggest that improved geometric initialization offers a promising path to scaling up autoregressive mesh generation across diverse shape distributions.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and improve the manuscript's readability. We did not use generative AI tools to generate research results, conduct experiments, or formulate scientific claims. All AI-assisted revisions have been reviewed and verified by the authors. We take responsibility for the final content of this study, including any text generated with the assistance of generative AI.

REFERENCES

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

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. 7, 20, 22

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, 8, 15

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22189–22199. IEEE, 2023. 3

Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Zhibin 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. 1, 3

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Zhongang Cai, Lei Yang, Gang Yu, Guosheng Lin, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. In *International Conference on Learning Representations*, volume 2025, pp. 51369–51389, 2025a. 3

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. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13922–13931. IEEE, 2025b. 3

Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5932–5941. IEEE, 2019. 3

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*, pp. 13142–13153, 2023. 1, 8, 15

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. 3

Xianglong He, Junyi Chen, Di Huang, Zexiang Liu, Xiaoshui Huang, Wanli Ouyang, Chun Yuan, and Yangguang Li. Meshcraft: Exploring efficient and controllable mesh generation with flow-based dits. *arXiv preprint arXiv:2503.23022*, 2025. 3

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 6, 20, 22

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 3

Jeonghwan Kim, Yushi Lan, Armando Fortes, Yongwei Chen, and Xingang Pan. Fastmesh: Efficient artistic mesh generation via component decoupling. In *2026 International Conference on 3D Vision (3DV)*, pp. 1478–1488. IEEE, 2026. 3

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2, 5, 21

Zeqiang Lai, Yunfei Zhao, Zibo Zhao, Haolin Liu, Qingxiang Lin, Jingwei Huang, Chunchao Guo, and Xiangyu Yue. Lattice: Democratize high-fidelity 3d generation at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19982–19992, 2026. 3

Jiabao Lei, Kewei Shi, Zhihao Liang, and Kui Jia. Armesh: Autoregressive mesh generation via next-level-of-detail prediction. *Advances in Neural Information Processing Systems*, 38:161613–161633, 2026. 1, 3

Haoxuan Li, Ziya Erkoç, Daniele Sirigatti, Vladislav Rosov, Lei Li, Angela Dai, and Matthias Nießner. Triflow: Generating artist-like 3d mesh topology via nearest-vertex vector fields. In *European Conference on Computer Vision*, pp. 407–425. Springer, 2026a. 3

Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025a. 3

Weiyu Li, Antoine Toisoul, Tom Monnier, Roman Shapovalov, Rakesh Ranjan, Ping Tan, and Andrea Vedaldi. Meshflow: Efficient artistic mesh generation via meshvae and flow-based diffusion transformer. *arXiv preprint arXiv:2606.04621*, 2026b. 3

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. 3

Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6526. IEEE, 2024. 3

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 300–309. IEEE, 2023. 3

Junkai Lin, Hang Long, Huipeng Guo, Jielei Zhang, JiaYi Yang, Tianle Guo, Yang Yang, Jianwen Li, Wenxiao Zhang, Matthias Nießner, et al. Meshripple: Structured autoregressive generation of artist-meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12706–12718, 2026. 3

Stefan Lionar, Jiabin Liang, and Gim Hee Lee. Treemeshgpt: Artistic mesh generation with autoregressive tree sequencing. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26608–26617. IEEE, 2025. 3

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 7, 20

Jian Liu, Haohan Weng, Biwen Lei, Xianghui Yang, Zibo Zhao, Zhuo Chen, Song Guo, Tao Han, and Chunchao Guo. Freemesh: Boosting mesh generation with coordinates merging. *arXiv preprint arXiv:2505.13573*, 2025. 3

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023. URL https://mlanthology.org/iclr/2023/liu2023iclr-flow/. 7, 20, 22

Hang Long, Tianhao Zhao, Junkai Lin, Youjia Zhang, Huipeng Guo, Rendong Liang, Jiale Xu, Jozef Hladkỳ, Matthias Nießner, Yuanming Hu, et al. Lato. 2: Factorized 3d mesh generation with vertex and topology flow. *arXiv preprint arXiv:2607.10623*, 2026. 3

Julieta Martinez, Holger H Hoos, and James J Little. Stacked quantizers for compositional vector compression. *arXiv preprint arXiv:1411.2173*, 2014. 19

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12663–12673. IEEE, 2023. 3

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. *Communications of the ACM*, 65(1):99–106, 2021. 3

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020. 1, 3, 15

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

Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9914–9925. IEEE, 2024. 3

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022. 2, 6

Tianchang Shen, Zhaoshuo Li, Marc Law, Matan Atzmon, Sanja Fidler, James Lucas, Jun Gao, and Nicholas Sharp. Spacemesh: A continuous representation for learning manifold surface meshes. In *SIGGRAPH Asia 2024 conference papers*, pp. 1–11, 2024. 3

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International conference on learning representations*, volume 2024, pp. 39838–39859, 2024. 3

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19615–19625. IEEE, 2024. 1, 3, 8, 9, 15

Qi Sun, Kiyohiro Nakayama, Jing Nathan Yan, Qixing Huang, Alexander Rush, Leonidas Guibas, Gordon Wetzstein, Jing Liao, and Guandao Yang. Meshflow: Mesh generation with equivariant flow matching. *arXiv preprint arXiv:2606.23489*, 2026. 3

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations*, volume 2024, pp. 33879–33896, 2024. 3

Jiaxiang Tang, Max Li, Zekun Hao, Xian Liu, Gang Zeng, Ming-Yu Liu, and Qinsheng Zhang. Edgerunner: Auto-regressive auto-encoder for artistic mesh generation. In *International Conference on Learning Representations*, volume 2025, pp. 35913–35934, 2025. 3

Hanxiao Wang, Yuan-Chen Guo, Ying-Tian Liu, Zi-Xin Zou, Biao Zhang, Weize Quan, Ding Liang, Yan-Pei Cao, and Dong-Ming Yan. Face: A face-based autoregressive representation for high-fidelity and efficient mesh generation. *arXiv preprint arXiv:2603.01515*, 2026a. 1, 3, 5, 6, 9, 15, 21

Hanxiao Wang, Ying-Tian Liu, Yuan-Chen Guo, Qi-Yuan Feng, Zi-Xin Zou, Ding Liang, Biao Zhang, and Yan-Pei Cao. Nexus: Native mesh generation with diffusion. *ACM Transactions on Graphics (TOG)*, 45(4):1–14, 2026b. 3

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12619–12629. IEEE, 2023a. 3

Yuxuan Wang, Xuanyu Yi, Haohan Weng, Qingshan Xu, Xiaokang Wei, Xianghui Yang, Chunchao Guo, Long Chen, and Hanwang Zhang. Nautilus: Locality-aware autoencoder for scalable mesh generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–10. IEEE, 2025. 3

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*, 36:8406–8441, 2023b. 3

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. 1, 3, 7, 8, 9, 15, 19, 22

Haohan Weng, Zibo Zhao, Biwen Lei, Xianghui Yang, Jian Liu, Zeqiang Lai, Zhuo Chen, Yuhong Liu, Jie Jiang, Chunchao Guo, et al. Scaling mesh generation via compressive tokenization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11093–11103, 2025. 3

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. 3

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2026. 3

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. 3

J Xiang et al. Trellis. 2: Native and compact structured latents for 3d generation, 2025a. 3

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025b. 3

Jiale Xu, Rendong Liang, Yuhao Long, Siyuan Shen, Zangyueyang Xian, Zeyi Xu, and Yuanming Hu. Meshy t2: Fast native mesh generation with flow matching. *arXiv preprint arXiv:2607.28675*, 2026a. 3

Jiale Xu, Wang Zhao, and Ying Shan. Meshweaver: Sparse-voxel-guided surface weaving for autoregressive mesh generation. *arXiv preprint arXiv:2606.04688*, 2026b. 3

Rui Xu, Tianyang Xue, Qiujie Dong, Le Wan, Zhe Zhu, Peng Li, Zhiyang Dou, Cheng Lin, Shiqing Xin, Yuan Liu, et al. Meshmosaic: scaling artist mesh generation via local-to-global assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20003–20013, 2026c. 3

Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4540–4549. IEEE, 2019. 3

Yichen Yang, Hong Li, Haodong Zhu, Linin Yang, Guojun Lei, Sheng Xu, and Baochang Zhang. Partdiffuser: Part-wise 3d mesh generation via discrete diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19940–19949, 2026. 3

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6796–6807. IEEE, 2024. 3

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions On Graphics (TOG)*, 43(4):1–20, 2024. 3

Ruowen Zhao, Junliang Ye, Zhengyi Wang, Guangce Liu, Yiwen Chen, Yikai Wang, and Jun Zhu. Deepmesh: Auto-regressive artist-mesh creation with reinforcement learning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10612–10623. IEEE, 2025a. 3

Tianhao Zhao, Youjia Zhang, Hang Long, Jinshen Zhang, Wenbing Li, Yang Yang, Gongbo Zhang, Jozef Hladkỳ, Matthias Nießner, and Wei Yang. Lato: 3d mesh flow matching with structured topology preserving latents. *arXiv preprint arXiv:2603.06357*, 2026. 3

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*, 36:73969–73982, 2023. 3

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025b. 3

Junzhe Zhu, Peiye Zhuang, and Sanmi Koyejo. Hifa: High-fidelity text-to-3d generation with advanced diffusion guidance. In *International Conference on Learning Representations*, volume 2024, pp. 5762–5786, 2024. 3

APPENDIX

In the appendix, we provide additional analysis in Section A, introduce seed-modeling variants in Section B, and report additional experimental results in Section C.

A FIRST-FACE DISTRIBUTION AND MODELING ANALYSIS

This section provides a more detailed analysis of the first-face modeling bottleneck introduced in Fig. 2. We examine the problem from three complementary perspectives: (i) how autoregressive modeling difficulty varies between the first and later faces, (ii) how complex the empirical first-face distribution is, and (iii) how well the autoregressive model fits this distribution. All first-face statistics are computed using equally sized subsets from ShapeNet (Chang et al., 2015) chair, ShapeNet table, and Objaverse (Deitke et al., 2023), with the same coordinate quantization, token ordering, and preprocessing used for model training.

A.1 FIRST FACE VS. LATER FACES

We first examine where the uncertainty of autoregressive mesh generation (Nash et al., 2020; Siddiqui et al., 2024; Weng et al., 2024; Wang et al., 2026a) is concentrated along the generation sequence. A face may either introduce previously unseen vertices or reuse vertices that have already appeared in the generated mesh. The first face is unique in that all of its vertices are novel and must be predicted without any preceding geometric context, whereas later faces increasingly benefit from previously generated geometry. We therefore separately analyze the cross-entropy of the first face, novel vertices in later faces, and reused vertices in later faces.

First-vs.-Later Cross-Entropy. For the comparison in Fig. 2a, we compute the cross-entropy of coordinate tokens under the ground-truth prefix and aggregate the results over first-face tokens, novel vertices in later faces, and reused vertices in later faces. Let $x_{i,k}$ denote the k-th coordinate token of the i-th mesh, and let $x_{i,<k}$ denote its preceding token sequence. For a set of evaluated sample-token pairs $\mathcal{I}$, the cross-entropy is computed as

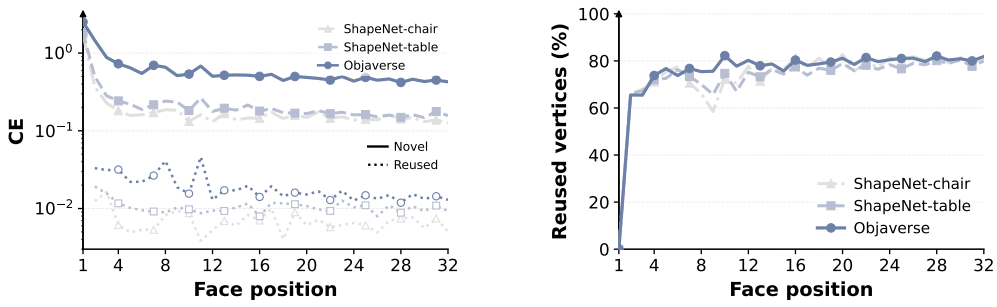
$$\mathrm{CE}_{\mathcal{I}} = -\frac{1}{|\mathcal{I}|} \sum_{(i,k)\in\mathcal{I}} \log p_\theta \left(x_{i,k} \mid x_{i,<k}\right), \tag{13}$$

where p_θ denotes the autoregressive mesh model evaluated under teacher forcing. We construct $\mathcal{I}$ separately for first-face tokens, coordinate tokens of novel vertices in later faces, and coordinate tokens associated with reused vertices. The first face exhibits the highest cross-entropy on all datasets, while reused vertices are substantially easier to predict.

Face-wise Cross-Entropy. We further resolve this comparison along the face sequence in Fig. 6a. For each face position t, we separately average the cross-entropy over novel and reused vertices in F_t. The first face contains only novel vertices and exhibits a sharp peak in uncertainty. After initialization, the cross-entropy of novel vertices decreases considerably, while that of reused vertices remains consistently low. Although Objaverse remains more difficult than ShapeNet throughout the sequence, the largest difference is concentrated at the beginning of generation.

Vertex Reuse Ratio. To quantify how much later faces reuse previously generated geometry, Fig. 6b reports the vertex reuse ratio at each face position. Let $N_{\mathrm{reuse}}(t)$ denote the number of vertices in F_t that have already appeared in preceding faces. Since each face contains three vertices, the reuse ratio is $R_t = N_{\mathrm{reuse}}(t)/3$. We then average R_t over all meshes at the same face position. While the first face necessarily has zero reuse, the reuse ratio rises rapidly after initialization and remains around 70–80% for most later positions. Consequently, much of later-face generation operates on geometry that has already been established rather than requiring entirely new vertex coordinates.

Together, these results show that autoregressive mesh modeling difficulty is strongly non-uniform across the face sequence. The first face simultaneously lacks preceding geometric context and requires entirely novel vertex prediction, whereas later faces increasingly operate within an already



(a) Cross-entropy (b) Vertex reuse

Figure 6: **Face-wise modeling difficulty.** (a) Cross-entropy of novel and reused vertices. The first face exhibits the highest uncertainty, while reused vertices in later faces are substantially easier to predict. (b) Ratio of reused vertices. After initialization, approximately 70–80% of vertices are reused from previously generated geometry. These results show that the first face uniquely requires prediction from no preceding geometry while introducing entirely novel vertices.

established geometric structure. This explains why the uncertainty observed in Fig. 2a is concentrated primarily at the first face and motivates treating initialization separately from subsequent causal generation.

A.2 First-Face Distribution Complexity

Having established that the first face is the most difficult stage of autoregressive generation, we next examine whether this difficulty originates from the first-face distribution itself. Each first face is represented by nine quantized coordinate tokens, $X_{1:9} = (X_1, \ldots, X_9)$, corresponding to the three coordinates of its three vertices. We characterize how the empirical distribution expands across these nine slots using prefix entropy, effective prefix count, conditional entropy, and effective branching factor.

Prefix Entropy. Fig. 7a reports the cumulative prefix entropy $\mathcal{H}(X_{1:k})$. Given the empirical distribution $\hat{p}(x_{1:k})$ over prefixes of length k, we compute the prefix entropy as

$$\mathcal{H}(X_{1:k}) = -\sum_{x_{1:k}} \hat{p}(x_{1:k}) \log \hat{p}(x_{1:k}). \tag{14}$$

This curve shows how uncertainty accumulates as the first face is progressively specified. Most of the entropy is accumulated within the earliest slots, after which the curve gradually saturates. The higher saturation level on Objaverse further indicates that it contains a broader and more diverse set of first-face configurations than the category-specific ShapeNet subsets.

Effective Prefix Count. The prefix-diversity analysis in Fig. 2b measures the effective number of distinct prefixes represented by the empirical distribution. Based on the prefix entropy defined above, we define the effective prefix count as

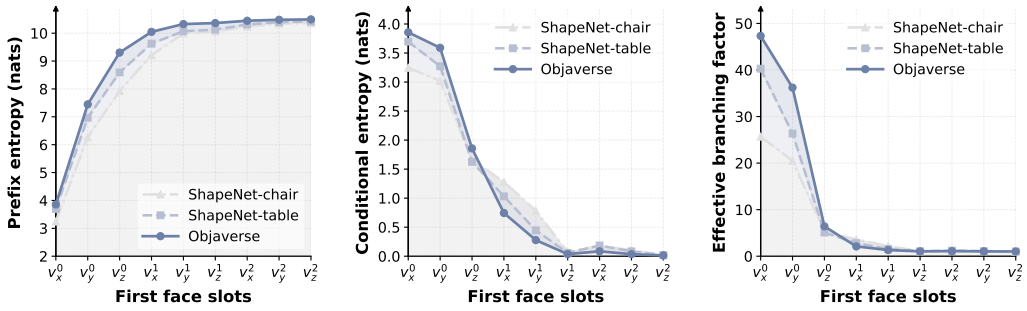
$$N_{\text{eff}}(k) = \exp(\mathcal{H}(X_{1:k}))\,. \tag{15}$$

Unlike the raw number of unique prefixes, $N_{\text{eff}}(k)$ discounts extremely rare observations and measures the effective support size of the prefix distribution. Objaverse exhibits a consistently larger effective prefix count, indicating a substantially broader first-face distribution.

Conditional Entropy. Fig. 7b reports the uncertainty introduced at each slot. The empirical conditional entropy is computed from the prefix entropy as

$$\mathcal{H}(X_k \mid X_{<k}) = \mathcal{H}(X_{1:k}) - \mathcal{H}(X_{1:k-1}), \tag{16}$$

with $\mathcal{H}(X_{1:0}) = 0$. Equivalently, $\mathcal{H}(X_{1:k}) = \sum_{j=1}^{k} \mathcal{H}(X_j \mid X_{<j})$. Conditional entropy measures the intrinsic uncertainty of each prediction in nats. The entropy is strongly front-loaded, where the



(a) Prefix entropy (b) Conditional entropy (c) Branching factor

Figure 7: **First-face distribution complexity.** (a) Prefix entropy accumulates rapidly over the earliest slots and then gradually saturates, showing that most first-face uncertainty is introduced near the beginning of the sequence. (b) Conditional entropy, corresponding to the incremental uncertainty contributed by each slot, is therefore strongly concentrated at the earliest positions. (c) The effective branching factor, obtained from the conditional entropy, shows the same trend and rapidly decreases as more coordinates are observed. Across all three measures, Objaverse exhibits greater first-face complexity than ShapeNet, particularly at the beginning of the sequence.

first several slots carry most of the uncertainty, while the final slots become nearly determined after enough coordinates of the face have been observed.

Effective Branching Factor. Fig. 7c examines how many plausible continuations remain at each slot. We convert the conditional entropy defined above into an effective branching factor,

$$B_{\text{eff}}(k) = \exp(\mathcal{H}(X_k \mid X_{<k}))\,. \tag{17}$$

This quantity can be interpreted as the effective number of plausible values for the next token after observing the current prefix. The branching factor is largest at the earliest coordinate slots and decreases rapidly as more of the first face becomes specified. Objaverse exhibits the largest early branching factors, showing that its first-face generation starts from a substantially less constrained state.

Overall, the earliest coordinates simultaneously admit many plausible continuations and contribute most of the total first-face entropy. This effect is considerably stronger on Objaverse, providing a data-distribution explanation for why first-face prediction becomes increasingly difficult as mesh diversity grows.

A.3 First-Face Modeling Gap and Empirical Coverage

The previous analysis characterizes the intrinsic uncertainty of the data distribution, but high data entropy alone does not imply that an autoregressive model fails to represent it. We therefore separate intrinsic uncertainty from model mismatch and further analyze how well finite samples cover the first-face distribution. This allows us to distinguish a genuinely diverse data distribution from additional modeling error introduced by the learned autoregressive predictor.

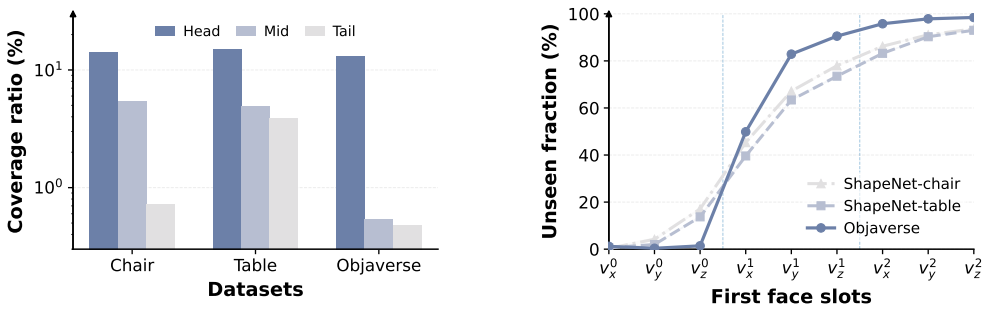
Modeling Gap. The modeling gap in Fig. 2c follows the definition in Eq. 1. At each first-face token position k, the empirical conditional entropy $\mathcal{H}(X_k \mid X_{<k})$ measures the intrinsic uncertainty of the data distribution after observing the ground-truth prefix. In contrast, the teacher-forced model cross-entropy CE_k measures the negative log-likelihood assigned by the autoregressive model to the ground-truth next token under the same prefix.

These two quantities are related by

$$\mathrm{CE}_k = \mathcal{H}(X_k \mid X_{<k}) + \mathbb{E}_{X_{<k}\sim p_{\text{data}}}\left[D_{\mathrm{KL}}\left(p_{\text{data}}(X_k \mid X_{<k}) \,\|\, p_\theta(X_k \mid X_{<k})\right)\right]. \tag{18}$$

Accordingly, the modeling gap

$$G_k = \mathrm{CE}_k - \mathcal{H}(X_k \mid X_{<k}) \tag{19}$$



(a) Coverage ratio (b) Unseen fraction

Figure 8: **Empirical coverage of the first-face distribution.** (a) Coverage ratio of the head, mid, and tail regions. Coverage is much higher in the head than in the mid and tail, especially on Objaverse. (b) Fraction of unseen first-face prefixes across token positions. The unseen fraction rises rapidly with prefix length, showing that coverage quickly becomes sparse for more specific prefixes.

corresponds to the expected conditional KL divergence between the empirical next-token distribution and the autoregressive model. It therefore removes the portion of prediction difficulty that is inherent to the data distribution and isolates the additional error arising from model–data mismatch. A high conditional entropy indicates that the next token is intrinsically difficult to predict, whereas a large G_k indicates that the model fails to capture the conditional distribution even after this intrinsic uncertainty is accounted for. Objaverse exhibits a consistently larger modeling gap, indicating that its first-face distribution is not only more uncertain but also more difficult for the autoregressive model to fit.

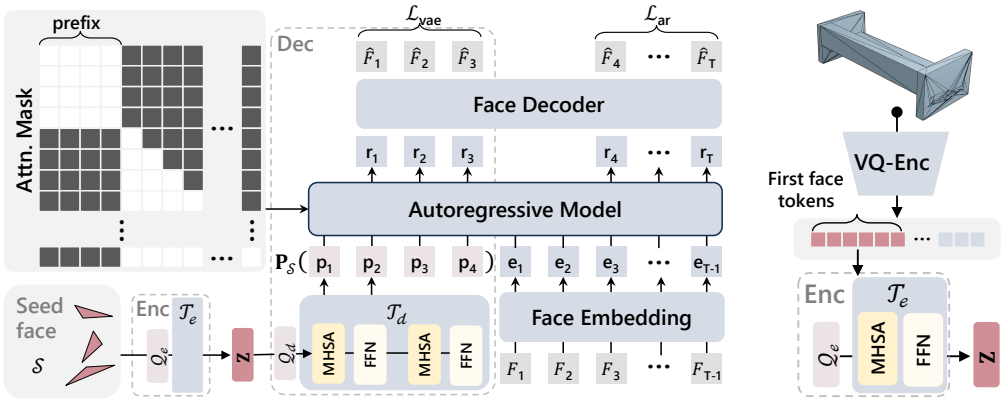
Coverage Ratio. Fig. 8a further examines support coverage at the complete first-face prefix depth. We divide the empirical first-face distribution into head, mid, and tail regions according to prefix frequency and compare the number of prefixes covered by model samples with that obtained from an equally sized resample of the data. For each region $\mathcal{R}$, the reported coverage ratio is $\mathrm{CR}_{\mathcal{R}} = |\mathcal{U}_{\mathrm{model}} \cap \mathcal{R}| / |\mathcal{U}_{\mathrm{resample}} \cap \mathcal{R}| \times 100\%$, where $\mathcal{U}_{\mathrm{model}}$ and $\mathcal{U}_{\mathrm{resample}}$ denote the sets of unique complete first-face prefixes obtained from the model samples and the equally sized data resample, respectively. Coverage decreases sharply from the head toward the tail, and this imbalance is particularly severe on Objaverse. Thus, the learned model captures frequent first-face configurations more readily than the long-tail portion of the distribution.

Unseen Fraction. Fig. 8b analyzes the same sparsity progressively across token positions. For each slot k, let $N_{\mathrm{unseen}}(k)$ denote the number of prefixes in the evaluation sample that are unseen in the sampled reference subset, and let $N_{\mathrm{eval}}(k)$ denote the total number of evaluated prefixes. The unseen fraction is $U_k = N_{\mathrm{unseen}}(k)/N_{\mathrm{eval}}(k) \times 100\%$. Short prefixes are well covered, but the unseen fraction rises rapidly as additional coordinates are included. After the first vertex is specified, a large fraction of longer prefixes are already unseen, and nearly complete first-face prefixes become overwhelmingly sparse. The increase is fastest on Objaverse, consistent with its substantially larger effective prefix space.

Taken together, these results show that the first-face bottleneck reflects both distributional complexity and model mismatch. Objaverse contains a broad, long-tailed first-face distribution whose support becomes sparse very quickly as the prefix grows, while the autoregressive model exhibits an additional cross-entropy gap beyond this intrinsic uncertainty. The combination of high early-stage diversity, sparse empirical coverage, and a larger modeling gap motivates explicitly learning the first-face distribution rather than relying solely on causal token-by-token initialization.

B SeedFACE Variants

This section examines several variants and extensions of SeedFACE. We first extend the formulation from a single seed face to multiple seed faces, where several initial faces are generated jointly be-



(a) Multi-face SeedFACE (b) PivotMesh adaptation

Figure 9: **Multi-face seed modeling and PivotMesh adaptation.** (a) SeedFACE with three seed faces ($T_s = 3$). The seed-query tokens are mutually visible, enabling the three seed faces to be generated in parallel. Once inserted into the mesh sequence, subsequent faces are generated using the standard causal autoregressive process. (b) Adaptation to PivotMesh. The first face is represented by discrete tokens from the pretrained VQ-Enc, and their codebook embeddings are encoded into the seed latent using the same query-based encoder. Seed-prefix decoding and subsequent causal generation then follow the same formulation as in (a).

fore causal autoregressive generation begins. We then adapt the same seed-face modeling strategy to PivotMesh (Weng et al., 2024), showing how the framework can accommodate a different discrete mesh representation. Finally, we introduce lightweight seed-face plugins, including diffusion, Rectified Flow, D3PM, and an isolated autoregressive model, which independently model the first-face distribution and can be combined with a pretrained autoregressive generator without retraining its subsequent generation process.

B.1 Multiple Seed Faces

While the main experiments use a single seed face by default, our formulation naturally extends to multiple seed faces. Here, we investigate how jointly modeling multiple initial faces affects subsequent autoregressive generation.

Let the seed consist of the first T_s faces, $\mathcal{S} = \mathcal{F}_{1:T_s} = (F_1, \ldots, F_{T_s})$. As illustrated in Fig. 9a, the seed latent $\mathbf{Z}$ is decoded into a prefix sequence $\mathbf{P}_{\mathcal{S}} = (\mathbf{p}_1, \ldots, \mathbf{p}_{N_d})$, where $N_d \geq T_s$. All prefix tokens are mutually visible, and the last T_s prefix positions serve as prediction slots for the seed faces. The corresponding seed-face representations are therefore produced in parallel from the shared prefix context, rather than autoregressively conditioned on one another.

After all seed faces are generated, they are inserted into the mesh sequence and used as the initial geometric context for subsequent generation. The standard causal structure is then restored: each later face can attend only to the seed faces and previously generated faces. Formally, causal generation starts from F_{T_s+1}. For the example in Fig. 9a, $T_s = 3$, so F_1, F_2, and F_3 are generated jointly from the seed prefix, while generation from F_4 onward follows the original autoregressive process. This extension changes only the number of seed prediction slots and leaves the subsequent autoregressive decoding unchanged.

B.2 Extension to PivotMesh

We further apply the proposed seed-face modeling strategy to PivotMesh (Weng et al., 2024), demonstrating that SeedFACE is not tied to a particular autoregressive mesh representation. Unlike our default model, PivotMesh represents the first face using discrete tokens obtained from a pretrained VQ encoder (Martinez et al., 2014). We therefore adapt only the seed encoder while preserving the remaining seed-prefix formulation.

As shown in Fig. 9b, the first-face tokens are first encoded by the pretrained VQ-Enc into discrete codebook indices. Their corresponding codebook embeddings are then treated as the input seed representation. A set of learnable query tokens attends to these embeddings through the seed encoder to extract the latent representation $\mathbf{Z}$. From this point onward, the procedure follows the same SeedFACE formulation: the latent is decoded into a seed prefix, which provides the initial context for the pretrained autoregressive mesh generator.

For PivotMesh, the first face is represented by six discrete tokens. Accordingly, the resulting seed prefix contains six seed positions, after which generation proceeds with the original causal autoregressive structure. Thus, adapting SeedFACE to PivotMesh requires modifying only the first-face encoding pathway, while retaining its pretrained autoregressive generation process.

B.3 Lightweight Seed-Face Plugin

We further explore a lightweight alternative that improves seed-face generation without retraining the underlying autoregressive model. The plugin independently models the seed-face distribution and uses the generated seed as the initial context for the pretrained autoregressive generator.

Diffusion. For the diffusion-based variant, we adopt a factorized process rather than directly modeling the nine coordinate values of the first face. Given its three vertices $\{\mathbf{v}_i\}_{i=1}^{3}$, we decompose the continuous geometry into a structured 13-dimensional state. Specifically, we compute the centroid $\mathbf{c} \in \mathbb{R}^3$, RMS scale s, and normalized local coordinates $\mathbf{u}_i = (\mathbf{v}_i - \mathbf{c})/s$, for $i \in \{1, 2, 3\}$. The resulting factorized representation is

$$\mathbf{y} = [\mathbf{c}, \log s, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] \in \mathbb{R}^{13}. \tag{20}$$

The centroid and log-scale are standardized using dataset statistics, while the normalized coordinates preserve the local triangle geometry. We further factorize $\mathbf{y}$ into semantic tokens corresponding to the centroid, scale, global triangle geometry, three edges, and three vertices, which are processed by a time-conditioned Transformer. At diffusion timestep τ, Gaussian noise $\boldsymbol{\epsilon}$ is added to obtain the noisy state $\mathbf{y}_{\mathrm{diff},\tau}$, and the denoising network predicts $\boldsymbol{\epsilon}$ using a standard DDPM objective (Ho et al., 2020). During sampling, the factorized state is progressively denoised from Gaussian noise and finally converted back to the vertex coordinates of the first face.

Rectified Flow. For the flow-based variant, we use the same factorized representation $\mathbf{y}$ and geometry-aware Transformer as in the diffusion variant, but replace the diffusion process with Rectified Flow (Liu et al., 2023; Lipman et al., 2022). Let $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ denote a Gaussian source sample. At continuous flow time $\rho \in [0, 1]$, we construct the linear interpolation

$$\mathbf{y}_{\mathrm{flow},\rho} = (1 - \rho)\boldsymbol{\xi} + \rho\mathbf{y}. \tag{21}$$

The Transformer predicts the target velocity $\mathbf{y} - \boldsymbol{\xi}$ from $\mathbf{y}_{\mathrm{flow},\rho}$ and ρ, and is trained with a mean-squared velocity loss. During inference, generation starts from Gaussian noise and integrates the learned velocity field from $\rho = 0$ to $\rho = 1$, after which the resulting factorized state is converted back to the vertex coordinates of the first face.

D3PM. D3PM (Austin et al., 2021) directly models the discrete first-face sequence $X_{1:9} = (X_1, \ldots, X_9)$, where each X_k is a quantized coordinate token. In the absorbing-state variant, an additional `MASK` state is introduced for the forward corruption process, while the clean output space remains the original coordinate vocabulary. At diffusion timestep τ, each clean token is either preserved or replaced by `MASK` according to the absorbing schedule, producing the corrupted sequence $X_{\mathrm{d3pm},\tau,1:9}$. A non-causal, time-conditioned Transformer jointly processes all nine slots using slot, vertex, and coordinate-axis embeddings, and predicts the clean sequence $X_{1:9}$. Training uses the absorbing-D3PM variational objective together with a small auxiliary clean-token cross-entropy term. Sampling starts from an all-`MASK` sequence and progressively reveals coordinate tokens until the complete first face is recovered.

Isolated AR. We additionally train an independent autoregressive model solely on the discrete first-face sequences. Unlike the full mesh autoregressive model, Isolated AR dedicates its entire modeling capacity to the nine coordinate tokens of the first face, without modeling any subsequent

Table 3: Comparison of different numbers of seed faces and ablation on Objaverse. COV, MMD, and 1-NNA are reported in %, 10^{-3}, and %, respectively. For 1-NNA, values closer to 50% are better. The best and second-best results among different T_s settings are highlighted in **bold** and underline, respectively. The setting without geometric features is included for reference only and is not considered in the ranking.

Metric	$T_s = 0$	$T_s = 1$	$T_s = 2$	$T_s = 5$	**w/o Geo. Feat.**
COV↑	44.34	46.68	<u>47.85</u>	**49.02**	46.09
MMD↓	18.00	**15.21**	<u>15.56</u>	16.23	15.71
1-NNA	69.34	**58.11**	<u>60.55</u>	63.87	59.42

geometry. This setup isolates the ability of a causal discrete model to fit the first-face distribution when its capacity is devoted exclusively to this task, and therefore serves as a discrete reference for comparison. Since the target consists of only a short sequence of quantized coordinates, Isolated AR can closely fit, or even largely memorize, the empirical discrete distribution rather than learning a scalable model of complete meshes.

C Extensive Experiments

This section presents additional experiments to further examine the flexibility and design choices of SeedFACE. We first investigate the effect of varying the number of seed faces, followed by an ablation of the geometric features used in the seed VAE encoder. We then evaluate the adaptation of SeedFACE to PivotMesh and different lightweight seed-modeling plugins for pretrained autoregressive models. Finally, we provide additional qualitative comparisons and generation results.

C.1 Number of Seed Faces

We first investigate the effect of modeling different numbers of seed faces. For this experiment, we construct a subset of 5000 meshes from the full training set and increase the capacity of the seed VAE (Kingma & Welling, 2013) to accommodate information from multiple faces. Specifically, both the query encoder and decoder are extended to four Transformer layers, the number of encoder queries is set to $N_e = 4$, and the latent dimensionality of each query is set to $C = 64$, resulting in $\mathbf{Z} \in \mathbb{R}^{N_e \times C} = \mathbb{R}^{4\times 64}$. We fix the number of decoded prefix tokens to $N_d = 6$ for all settings. When $T_s < N_d$, the remaining prefix tokens are retained as context tokens and are not assigned seed-face prediction targets. We use the same FACE (Wang et al., 2026a) and diffusion configurations and train all variants for the same number of epochs.

Table 3 reports the results for different values of T_s. Introducing seed-face modeling already yields a clear improvement over $T_s = 0$. Increasing T_s further improves coverage, with COV increasing from 44.34% at $T_s = 0$ to 49.02% at $T_s = 5$. However, MMD and 1-NNA are best at $T_s = 1$, while larger T_s slightly degrade these two metrics. This suggests that modeling more seed faces can broaden the support of generated shapes, but increasing the amount of jointly modeled seed geometry does not necessarily lead to monotonically better distribution matching. Overall, a small number of seed faces already captures most of the benefit, while additional seed faces mainly improve generation coverage.

C.2 VAE Encoder Ablation

We further examine the contribution of geometric features in the seed VAE encoder. Following the same experimental protocol in Sec. C.1, we compare the default $T_s = 1$ setting with a variant that removes the additional geometric features and uses only the coordinates of the three vertices of the first face as input. As shown in Table 3, removing geometric features results in only a modest performance change: COV decreases from 46.68% to 46.09%, MMD increases from 15.21 to 15.71, and 1-NNA changes from 58.11% to 59.42%. The relatively small gap suggests that, when modeling only a single seed face, its vertex coordinates already provide most of the information required by the latent encoder. The additional geometric features therefore serve mainly as auxiliary cues rather than being essential to first-face modeling.

Table 4: Comparison of different seed-face modeling strategies on Objaverse. COV, MMD, and 1-NNA are reported in %, 10^{-3}, and %, respectively. For 1-NNA, values closer to 50% are better. The best and second-best results are highlighted in **bold** and underline, respectively.

Metric	PivotMesh+Seed	PivotMesh	D3PM	AR	Diff	Flow	SeedFACE
COV↑	45.85	38.67	48.93	<u>52.59</u>	52.00	51.71	**52.88**
MMD↓	12.90	12.28	10.98	**10.43**	10.63	10.99	<u>10.46</u>
1-NNA	65.70	62.79	54.52	<u>54.17</u>	54.96	58.62	**53.56**

C.3 PivotMesh with Seed Face and Lightweight Plugin

We next evaluate the lightweight seed-modeling plugins that can be attached to a pretrained autoregressive mesh generator without retraining its subsequent generation process. We first consider the PivotMesh (Weng et al., 2024) adaptation described in Sec. B.2. As shown in Table 4, introducing the seed module increases the COV of PivotMesh from 38.67% to 45.85%, indicating that explicit modeling of the initial distribution enables the pretrained generator to reach a broader range of shapes. At the same time, MMD increases from 12.28 to 12.90, while 1-NNA moves from 62.79% to 65.70%, showing that the additional diversity is accompanied by reduced distribution fidelity.

One possible reason is the more difficult first-face representation used by PivotMesh. Instead of directly modeling the coordinates of three vertices, its first face is represented through discrete tokens from a codebook containing 16,384 entries. The seed module must therefore model and reconstruct a considerably larger discrete representation space before the pretrained autoregressive decoder can continue generation. Errors in this initial representation can propagate to subsequent faces, making generation less stable despite the increased coverage.

We also evaluate several lightweight first-face generators for autoregressive models that directly represent a face using three vertex coordinates. As shown in Table 4, discrete diffusion (Austin et al., 2021), isolated autoregressive modeling, continuous diffusion (Ho et al., 2020), and rectified flow (Liu et al., 2023) can all provide effective seed distributions before handing generation back to the pretrained autoregressive model. These results further support that explicitly separating first-face modeling from subsequent causal generation is applicable across different seed-generation mechanisms rather than depending on a particular architecture.

Interestingly, different seed models learn slightly different first-face distributions and consequently exhibit distinct generation tendencies. We visualize representative examples in the plugin generation gallery, where different plugins produce characteristic geometric structures even when followed by the same autoregressive generation process.

C.4 Qualitative Results

Fig. 10 collects representative meshes generated by the PivotMesh + SeedFACE variant and different lightweight seed-modeling plugins. We specifically select samples that exhibit characteristic geometric structures induced by their learned first-face distributions, illustrating that different seed models can lead to distinct generation tendencies even when followed by the same autoregressive generation process.

We then provide additional qualitative comparisons on both Objaverse and ShapeNet. Results on Objaverse are shown in Fig. 11, while category-wise comparisons on ShapeNet are presented in Fig. 12, Fig. 13, Fig. 14, and Fig. 15. These results further illustrate the differences among the compared methods across diverse object structures and categories.

Finally, we provide a larger gallery of samples generated by SeedFACE on Objaverse in Fig. 16, demonstrating the diversity of generated geometries over a broader range of object types.

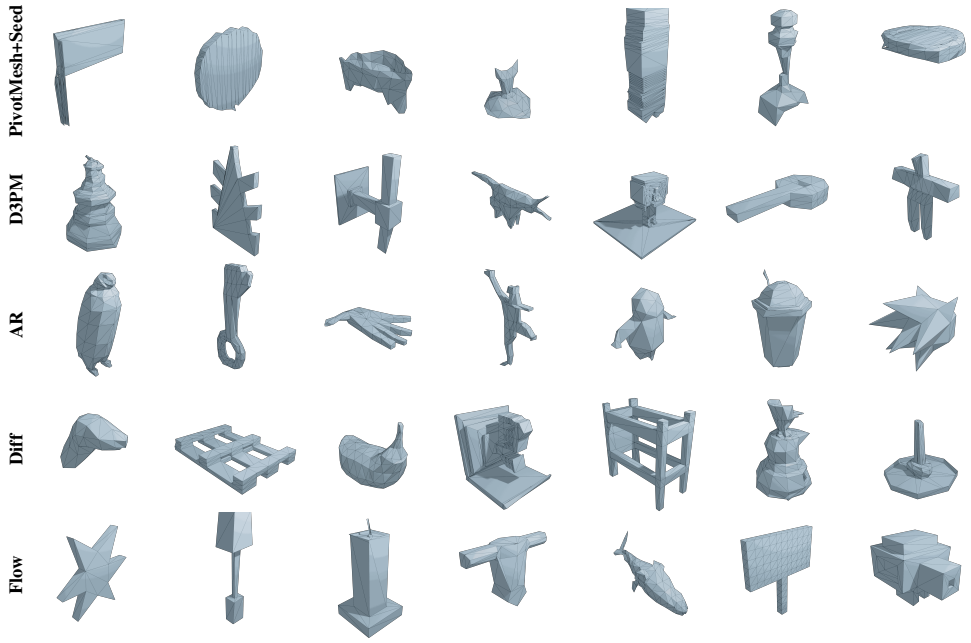


Figure 10: Generation gallery of novel objects by the lightweight plugins on Objaverse.



Figure 11: Additional qualitative comparison on Objaverse.

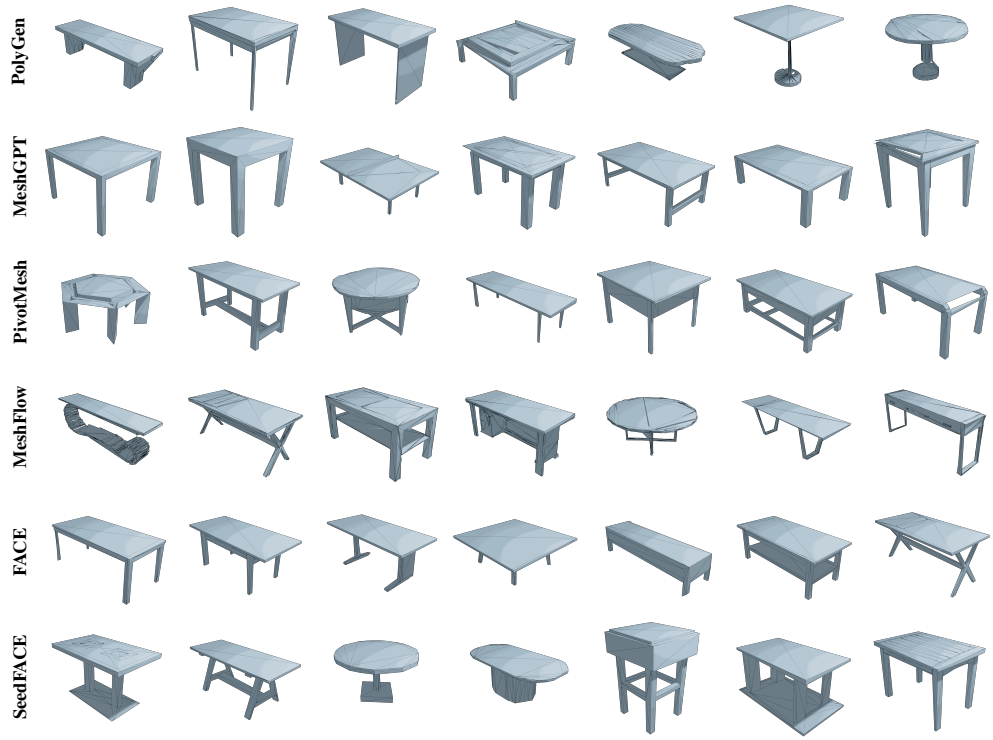


Figure 12: Additional qualitative comparison on ShapeNet for the **table** category.

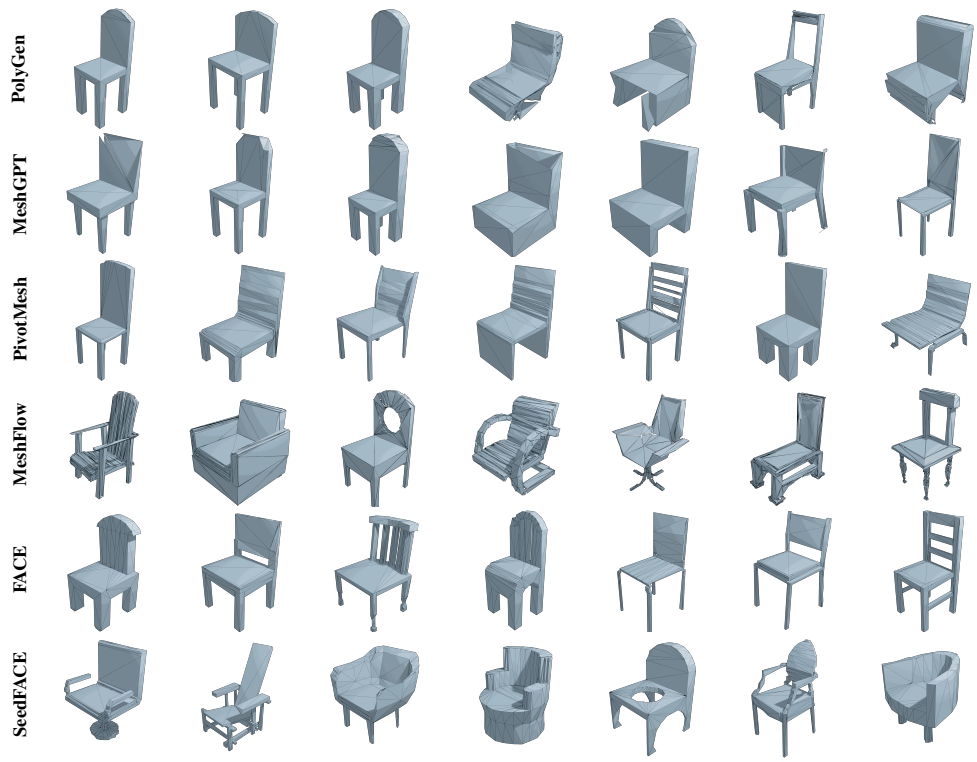


Figure 13: Additional qualitative comparison on ShapeNet for the **chair** category.

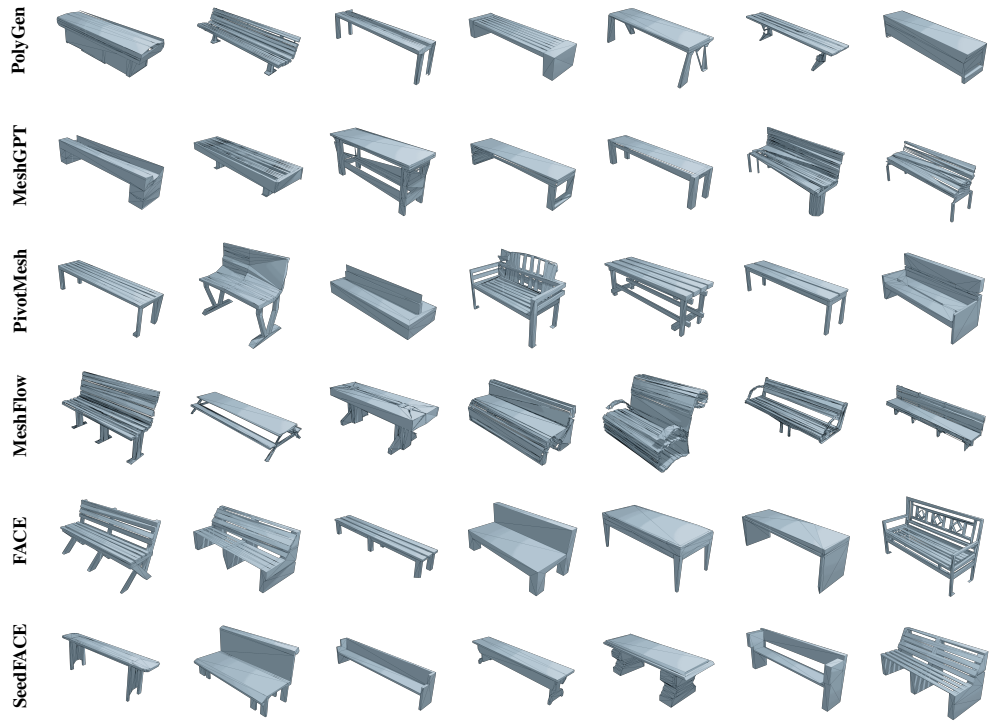


Figure 14: Additional qualitative comparison on ShapeNet for the **bench** category.

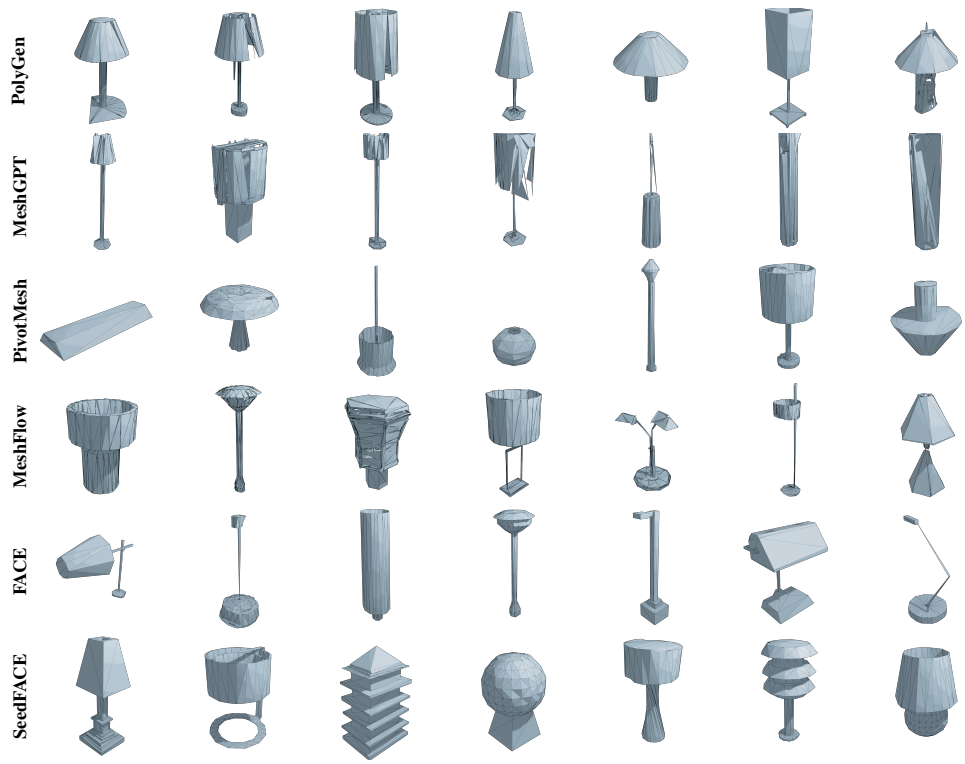


Figure 15: Additional qualitative comparison on ShapeNet for the **lamp** category.



Figure 16: Additional generation gallery of SeedFACE on Objaverse.