MAR-3D: Progressive Masked Auto-regressor for High-Resolution 3D Generation

Jinnan Chen[1]   Lingting Zhu[2]   Zeyu Hu[3]   Shengju Qian[3]
Yugang Chen[3]   Xin Wang[3]   Gim Hee Lee[1]

[1]National University of Singapore
[2]The University of Hong Kong
[3]LIGHTSPEED

https://jinnan-chen.github.io/projects/MAR-3D/

Figure 1. Our MAR-3D demonstrates strong generalization ability on in the wild images, accurately handling complex geometric details including fine structures and intricate shapes. We visualize the normal maps of our generated meshes with some random rotations.

Abstract

Recent advances in auto-regressive transformers have revolutionized generative modeling across different domains, from language processing to visual generation, demonstrating remarkable capabilities. However, applying these advances to 3D generation presents three key challenges: the unordered nature of 3D data conflicts with sequential next-token prediction paradigm, conventional vector quantization approaches incur substantial compression loss when applied to 3D meshes, and the lack of efficient scaling strategies for higher resolution latent prediction. To address these challenges, we introduce MAR-3D, which integrates a pyramid variational autoencoder with a cascaded masked auto-regressive transformer (Cascaded MAR) for progressive latent upscaling in the continuous space. Our architecture employs random masking during training and auto-regressive denoising in random order during inference, naturally accommodating the unordered property of 3D latent tokens. Additionally, we propose a cascaded training strategy with condition augmentation that enables efficiently up-scale the latent token resolution with fast convergence. Extensive experiments demonstrate that MAR-3D not only achieves superior performance and generalization capabilities compared to existing methods but also exhibits enhanced scaling capabilities compared to joint distribu-

tion modeling approaches (e.g., diffusion transformers).

1. Introduction

High-quality 3D mesh generation has emerged as a critical challenge in computer graphics and vision, with widespread applications in gaming and AR/VR industries. Recent advances in open-world 3D object generation have demonstrated promising results following several distinct paradigms. Large Reconstruction Models [13, 39] employ transformers to convert single images into 3D shapes through implicit representations with multi-view supervision. However, the lack of generative priors leads to blurry artifacts in unseen regions. Another direction combines 2D diffusion models [4, 21, 23, 33, 41] with sparse view 3D reconstruction models [18, 37, 46, 48]. However, these methods are constrained by the quality and consistency of the generated multi-view images. A more recent paradigm follows the success of 2D image generation by utilizing 3D variational auto-encoders (VAE) to compress dense point clouds into latent space, then applying diffusion models for direct 3D shape generation [14, 20, 50–52].

While these approaches show promising performance, they face fundamental challenges in effectively increasing tokens to achieve higher quality generation: (1) existing VAEs and generators struggle to maintain geometric detail when representing 3D data into limited number of tokens, (2) the computational complexity of transformer-based generators grows quadratically with token resolution, directly increasing the number of tokens requires hundreds of GPUs [51]—making it impractical, and (3) the lack of an effective strategy to progressively increase resolution while maintaining generation quality.

In this work, we address these challenges through a progressive approach. First, we introduce a Pyramid VAE that captures multi-scale geometric information through different cross-attention layers, improving data representation and reconstruction quality while maintaining efficient token resolution. Second, we develop a cascaded generation strategy using two Masked Auto-Regressive (MAR) models: MAR-LR generates low-resolution tokens that capture overall shape, while MAR-HR refines these into high-resolution tokens with fine geometric details.

This progressive strategy is enabled by several key strategies: (1) a random masking operation during training that naturally fits the unordered property of 3D latent tokens, (2) condition augmentation that reduces compounding errors when increasing resolution, and (3) an efficient parallel decoding strategy that maintains generation quality even with longer token sequences. Through extensive experiments, we demonstrate that our approach not only achieves superior performance compared to existing methods but also exhibits enhanced progressive properties in terms of both la-

tent token resolution and generation quality.

Our key contributions include:
- A Pyramid VAE architecture that enables effectively increasing token resolution to preserve geometric details.
- A cascaded MAR generation that combines MAR-LR and MAR-HR models for progressive token generation.
- A comprehensive study demonstrating the benefits of our progressive up-scaling strategy combining auto-regressive model for improving generation quality.
- The state-of-the-art results among open-sourced methods on public benchmarks, with particular improvements in preserving fine geometric details and complex structures.

2. Related works

2.1. 3D Large Reconstruction Models

The Large Reconstruction Model (LRM) [13, 39, 43, 45] marks a pivotal advancement in single-view 3D reconstruction by significantly scaling up both model architecture and dataset size for neural radiance field [25] (NeRF) prediction. While LRM was initially developed for reconstruction tasks, its capabilities have been extended to text-to-3D and image-to-3D generation through integration with multiview diffusion models [23, 32, 41, 42], as demonstrated by subsequent works such as Instant3D [18], DMV3D [47], and InstantMesh [46]. Recent innovations, including LGM [37], GRM [48], and LaRa [3], have further enhanced rendering quality and computational efficiency by combining novel view generation diffusion models with generalizable Gaussians family [15, 17] in a feed-forward manner. However, this approach of leveraging pre-trained multiview diffusion models [33, 41] encounters two significant challenges. First, the disjoint training of reconstruction and diffusion models can introduce multiview inconsistencies during inference. Second, the multiview diffusion process typically produces lower resolution outputs, resulting in the loss of original image details and degraded reconstruction fidelity.

2.2. 3D Generative Models

Direct generation of 3D content under explicit 3D supervision offers a more efficient approach to 3D content creation [16, 31, 49–53]. However, training generative models directly on 3D data poses significant challenges due to extensive memory and computational requirements. Recent approaches address these limitations by first compressing 3D shapes into compact latent representations before applying diffusion or auto-regressive models. This field saw significant advancement with 3DShape2VecSet [50], which introduced a 3D mesh VAE that encodes point clouds into shape latents and decodes occupancy values using grid embeddings, coupled with a categorical latent diffusion model. Building on this foundation, subsequent works including Michelangelo [52], CLAY [51], and Craftsman [20] scaled

up 3D diffusion models by leveraging diffusion transformer architectures and larger datasets, achieving superior generalization capabilities. Xcubes [31] further advanced the field by compressing 3D meshes into sparse latent voxels and learning hierarchical voxel diffusion models for generation. While recent attempts aiming at synthesizing meshes directly [5, 6, 26, 34] have demonstrated promising results in high-quality mesh generation, their effectiveness is limited by mesh tokenization constraints and strict requirements on face counts, which ultimately restrict their generalization capability. Distinct from previous approaches, our method preserves the advantages of auto-regressive models while circumventing the limitations of lossy tokenization from vector quantization through the integration of diffusion models and progressive up-scaling strategy.

2.3. Generative Auto-Regressive Models

Auto-regressive models [9, 40] have revolutionized visual generation [1, 2, 8, 19, 22, 27, 36, 38] through their sequential approach of synthesizing images using discrete tokens produced by image tokenizers. Pioneering works such as VQGAN [8] demonstrated the effectiveness of raster-scan sequences for next-token prediction by first training a discrete-valued tokenizer on images, utilizing a finite vocabulary obtained through vector quantization, followed by per-token prediction. Subsequently, Maskgit and MUSE [1, 2] advanced beyond sequential token prediction by introducing parallel prediction of multiple tokens in random order, substantially improving both generation quality and efficiency. More recently, VAR [38] introduced a novel next-scale prediction paradigm that not only better preserves spatial locality but also achieves significant computational efficiency gains. However, these methods still suffer from information loss due to quantization in the latent space. To address this limitation in continuous domains, MAR [19] proposed modeling per-token probability distributions using a diffusion process [11], effectively combining the efficiency of auto-regressive models with the advantages of continuous diffusion processes to mitigate quantization losses. Building upon these advancements and the demonstrated scalability benefits in 2D applications, our work extends continuous auto-regressive modeling into the realm of high quality 3D mesh generation.

3. Approach

3.1. Overall

As illustrated in Fig. 2, our framework MAR-3D consists of a Pyramid VAE architecture coupled with a Cascaded MAR. Specifically, MAR-LR generates low-resolution tokens encoding global structure conditioned on the input image tokens, while MAR-HR produces high-resolution tokens conditioned on the previously generated

low-resolution tokens and image tokens, enabling fine geometric detail refinement. The final 3D mesh is extracted by applying Marching Cubes to the occupancy field [24] from the high-resolution latent tokens.

3.2. Pyramid VAE

Encoder As illustrated in Fig. 2 (a), our Pyramid VAE first generates K levels of down-sampled resolutions from the input point cloud $\mathbf{P}^k \in \mathbb{R}^{N^k \times 3}$, where N^k is the number of points for k_{th} level, and concatenates them with their corresponding surface normals $\mathbf{P}_n^k$ as multi-resolution point embeddings:

$$\hat{\mathbf{P}}^k = [\gamma(\mathbf{P}^k), \mathbf{P}_n^k], \tag{1}$$

where $\gamma(\cdot)$ denotes the positional embedding function. Subsequently, we apply cross-attention operations between learnable queries $\mathbf{S}$ and each level of point embedding $\hat{\mathbf{P}}^k$, which serve as keys and values. Each resolution level employs distinct cross-attention layers: coarse levels capture structural features while fine levels extract detailed geometric information, which are then added together. Multiple self-attention layers are subsequently applied to obtain the latent tokens $\mathbf{X}$:

$$\mathbf{X} = \text{SelfAttn}\left(\sum_{k=1}^{K} \text{CrossAttn}^k(\mathbf{S}, \hat{\mathbf{P}}^k)\right). \tag{2}$$

While directly using high-resolution latent tokens would increase computational burden and complicate diffusion training, our hierarchical design enables efficient token length compression while preserving geometric details. Unlike [51], which randomly operates on one of multiple resolutions of point clouds during training, our Pyramid VAE processes multiple resolution levels simultaneously and also supports latent tokens of different resolutions, which enables progressive scaling in the next step.

Decoder The decoder architecture first processes the encoded features through multiple self-attention layers. We then sample query points with positional embeddings and employ cross-attention between these embedded points and the processed latent tokens to predict occupancy values.

Training Objective We optimize the VAE using a combination of binary cross-entropy (BCE) loss for occupancy prediction and KL-divergence loss for latent space regularization:

$$\mathcal{L}_{\text{vae}} = \mathbb{E}_{x \in \mathbb{R}^3}\left[\text{BCE}\left(\hat{\mathcal{O}}(x), \mathcal{D}(\gamma(x), \mathbf{X})\right)\right] + \lambda_{\text{kl}}\mathcal{L}_{\text{kl}}, \tag{3}$$

where $\hat{\mathcal{O}}(x)$ is the ground truth occupancy value.

Figure 2. **Overview of MAR-3D:** (a) Pyramid VAE: It processes learnable tokens through separate cross-attention layers, taking multi-resolution point clouds and normals as input to generate occupancy fields. (b) Cascaded MAR: Conditioned on image features extracted by CLIP and DINOv2, we employ a cascaded design: a MAR-LR model for generating low-resolution tokens, and a MAR-HR model for high-resolution token. The MAR architecture details are illustrated in the blue box. While MAR-LR and MAR-HR share the same architecture, they differ in the inputs: MAR-HR additionally requires low-resolution tokens as input (shown in the dashed box).

3.3. Cascaded Masked Auto-Regressive Model

Our Cascaded MAR consists of MAR-LR and MAR-HR models with the same architecture while different input tokens. MAR-LR takes image tokens as condition and MAR-HR takes both of the image tokens and the low-resolution tokens generated by MAR-LR. Traditional auto-regressive models predict tokens sequentially based on previous tokens, following a causal ordering paradigm widely adopted in language models. However, given the unordered nature of our latent tokens, we employ a random and parallel decoding strategy inspired by image generation techniques [1]

to efficiently generate large number of tokens:

$$
\begin{aligned}
p(x^1, \ldots, x^N) &= p(X^1, \ldots, X^S) \\
&= \prod_S p(X^s | X^1, \ldots, X^{s-1}),
\end{aligned} \quad (4)
$$

where $\mathbf{X}^s$ represents a set of tokens generated in parallel at step s, the generation order of latent tokens is determined randomly. During training, we adopt Masked Autoencoders (MAE) [9], which employs random masking to reconstruct masked regions using information from unmasked tokens. This enables the model to predict tokens in arbitrary order during inference. Since our latent tokens exist in continuous space, we apply diffusion loss [19] rather than conventional Cross Entropy loss to supervise both of our auto-regressive model and diffusion model. During inference,

starting from image tokens, the newly generated tokens are positioned in their designated locations and iteratively fed back into the MAE encoder-decoder and diffusion denoising pipeline for subsequent continuous token generation. The low-resolution tokens generated by MAR-LR are concatenated with image tokens and fed into MAR-HR, following the same process to generate final high-resolution tokens, which are fed into our VAE decoder to generate the occupancy field and extract the mesh.

3.3.1. MAR-LR

Positional Embedding   Given that our latent tokens lack inherent sequential order or spatial position information, conventional positional embedding approaches such as absolute positional encoding and relative positional encoding are not applicable. Instead, we associate each latent token with learnable positional tokens, which are added as residuals to the original latent tokens, enabling positional embedding adaptively updated during training.

Image Tokens   For processing conditional images, we leverage complementary features from CLIP [30] and DINOv2 [28]. The concatenated features serve as initial tokens for the MAR encoder, providing both semantic understanding and fine-grained pixel-level features. During training, we implement classifier-free guidance by randomly nullifying conditional input features with 0.1 probability, enhancing conditional generation quality [10].

MAE Encoder and Decoder   We concatenate images tokens with latent tokens and apply random masking with a ratio ranging from 0.7 to 1.0. Following the MAE architecture, we employ bidirectional attention mechanisms. The process involves first processing unmasked tokens through the MAE encoder, which is composed of multiple self-attention layers, and then concatenating the encoded tokens with mask tokens—using the same pre-defined learnable token for all masked positions. Learnable positional embeddings are incorporated into both masked and unmasked tokens before entering the MAE decoder, enabling position-aware token prediction.

Diffusion process   The decoder generates each condition vector $\mathbf{z} \in \mathbb{R}^D$ for each token used in the diffusion process. A lightweight MLP-based denoising network then reconstructs ground truth tokens from Gaussian noise by optimizing:

$$\mathcal{L}(z, x) = \mathbb{E}z, t \left[|\epsilon - \epsilon\theta(x^t|t, z)|^2 \right], \quad (5)$$

where z is the condition vector from MAE decoder and x_t is the ground truth latent token provided by well-trained VAE. During inference, reverse diffusion process is applied to predict each set of tokens.

3.3.2. MAR-HR

We analyze the relationship between VAE latent token length and reconstruction error. While increasing token length improves VAE reconstruction quality, directly training MAR on longer sequences poses convergence challenges due to quadratic computational complexity. To achieve high-quality geometric details while maintaining computational efficiency, we implement a coarse-to-fine generation strategy using a super-resolution model (MAR-HR) that shares MAR-LR's architecture but generates high-resolution tokens conditioned on low-resolution latent tokens and image tokens. The training objective for MAR-HR is defined as:

$$\mathcal{L}_H(z_h, x_h) = \mathbb{E}z_h, t \left[|\epsilon - \epsilon_\theta(x_h^t|t, z_h)|^2 \right], \quad (6)$$

where x_h represents high-resolution tokens from the VAE and z_h denotes the condition vector from MAE decoder in our MAR-HR model.

Condition Augmentation   To address the discrepancy between VAE-generated low-resolution tokens used during training and MAR-LR-generated tokens used during inference, we employ a condition augmentation strategy to mitigate compounding error [12]. Our approach applies Gaussian noise to the low-resolution tokens x_l through:

$$x_l' = t\epsilon + (1 - t)x_l, \quad (7)$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and $t \sim \mathcal{U}(0.4, 0.6)$ during training, with t fixed at 0.5 during inference. This augmentation is applied to both VAE-generated low-resolution tokens during training and MAR-LR-generated low-resolution tokens during inference before they are processed by MAR-HR. This strategy effectively reduces compounding error and provides strong conditional guidance to MAR-HR, enabling faster convergence compared to direct high-resolution token training. The previous approach, as seen in [51], requires substantial computational resources (hundreds of GPUs) and, as demonstrated in our experiments, yields poor convergence under the same limited GPU resources.

3.4. Inference Schedules

Generation Schedule   During inference, we first generate a random token generation order. Then we extract image tokens from input image, which are fed into the MAE encoder-decoder architecture. From the decoder outputs, we select condition vector z according to the predetermined generation order. Multiple tokens undergo parallel DDIM [35] denoising processes simultaneously. The number of tokens N_s processed in each auto-regressive step follows a cosine schedule as in [1], progressively increasing over S total steps:

$$N_s = \left\lfloor N(cos(\frac{s}{S}) - cos(\frac{s-1}{S})) \right\rfloor. \quad (8)$$

Method	GSO [7]			OmniObject3D [44]		
	F-Score ↑	CD ↓	NC ↑	F-Score ↑	CD ↓	NC↑
LGM [37]	0.745	0.813	0.685	0.738	0.821	0.677
CraftsMan [20]	0.776	0.785	0.687	0.771	0.798	0.675
TripoSR [39]	0.834	0.644	0.727	0.825	0.621	0.731
InstantMesh [46]	0.923	0.415	0.780	0.918	0.427	0.779
Ours	**0.944**	**0.351**	**0.835**	**0.931**	**0.364**	**0.826**

Table 1. **Comparison of different methods on GSO and OmniObject3D datasets.** Arrows (↑/↓) indicate whether higher or lower is better. Best results are in **bold**.

This scheduling strategy is motivated by the observation that initial tokens are more challenging to predict, while later tokens become progressively easier to determine, similar to a completion task. Consequently, we generate fewer tokens in initial steps and gradually increase the number in later steps, rather than maintaining a constant generation rate across all steps.

CFG Schedule   We employ CFG in our diffusion model:

$$\varepsilon = \varepsilon_\theta(x_t|t, z_u) + \omega_s \cdot (\varepsilon_\theta(x_t|t, z_c) - \varepsilon_\theta(x_t|t, z_u)), \quad (9)$$

where z_c and z_u are conditional and unconditional output from the MAE decoder, which serve as the condition for the diffusion model. We employ a linear strategy [19] for the CFG coefficient ω_s, starting with lower values during the initial uncertain steps and progressively increasing it to λ_{cfg}. Specifically, in Eq. 9, we set $\omega_s = s * \lambda_{cfg}/S$.

3.5. Discussion

The MAR-3D architecture offers significant advantages over joint distribution modeling methods such as DiT by decomposing the complex joint distribution into temporal (diffusion) and spatial (auto-regressive) components. This decomposition, combined with our cascaded super resolution model, enables progressively increasing tokens. The effectiveness of this approach is comprehensively validated through ablation studies comparing against existing joint distribution modeling method by DiT [29].

4. Experiments

4.1. Implementation Details

We train our pyramid VAE using a hierarchical point cloud representation with 16384, 4096, and 1024 points from the highest to lowest level, along with 20480 sampled ground truth occupancy values. For cascaded MAR model training, we use the conditional images and latent tokens sampled from the pyramid VAE, with batch sizes of 32 and 8 per GPU for MAR-LR and MAR-HR respectively. Training and inference details are provided in the supplementary.

4.2. Training Strategies

Training Data Curation   Our training procedure utilizes carefully curated data from Objaverse [44], implementing a two-stage approach.We first train on 260K geometrically diverse meshes with partial low texture quality for 200 epochs, followed by fine-tuning on 60K high-quality meshes with natural textures for 100 epochs.

Rotation Augmentation   For each 3D mesh, we render 56 conditional views using a structured rendering approach. We first uniformly sample 8 base views with azimuth angles ranging from 0° to 360°. Each base view is then augmented with 6 different random rotations. To ensure consistency between the input images and 3D latent tokens, we apply corresponding rotations to the 3D meshes before VAE encoding. This strategy effectively avoids the ambiguity that would arise from mapping the same canonical mesh to different conditional view images.

4.3. Comparison Results

Evaluation Settings   We compare our method with recent single-view approaches: InstantMesh [46] (multiview diffusion with feed-forward SDF), LGM [37] (feed-forward 3D Gaussians), TripoSR [39] (large reconstruction model without diffusion), and CraftsMan [20] (VAE-diffusion pipeline similar to ours). All evaluations use official pre-trained models. We evaluate on GSO [7] and OmniObject3D [44] datasets, which contain unseen real-scanned objects. To ensure meaningful comparison, we first remove too simple categories such as boxes and then randomly sample 100 shapes from each dataset (200 total). We assess performance using F-score, Chamfer distance (CD), and Normal Consistency (NC) between predicted and ground truth meshes, after normalization and ICP alignment.

Quantitative Evaluation   The quantitative comparison with existing single-view mesh reconstruction and generation methods is presented in Tab. 1. Our method significantly outperforms baseline approaches across all evaluation metrics on both test datasets, demonstrating strong generalization to unseen data. On the GSO dataset, our

Figure 3. **Comparison on rendered normal map.** We visualize the normal map rendered by our method and other baseline methods.

method achieves a Chamfer Distance (CD) of 0.351, reducing the geometric error by 15.4% compared to InstantMesh (0.415) and showing even more substantial improvements over naive diffusion methods like CraftsMan (0.785). This significant reduction in CD indicates more accurate geometric reconstructions. The improvement is consistent on the OmniObject3D dataset, where our method maintains a low CD of 0.364, outperforming InstantMesh (0.427) and other competitors by a large margin. Additionally, we achieve strong performance in other metrics, with F-Score reaching 0.944 and Normal Consistency (NC) of 0.835 on GSO, further validating the effectiveness of our approach.

Qualitative Evaluation We conduct qualitative evaluation in Fig. 3 and showcase representative examples demonstrating our method's capabilities in handling challenging

cases, including multi-object scenes, intricate geometric structures, and meshes with topological holes. We visualize the normal map rendered from the meshes generated by these methods. As demonstrated in the figure, our method exhibits superior reconstruction capabilities compared to existing approaches. LGM struggles with geometric accuracy due to multi-view inconsistency and challenges in converting 3D Gaussians to high-quality meshes. While CraftsMan employs a similar VAE-diffusion pipeline, its naive design leads to incorrect geometry and incomplete reconstructions. InstantMesh produces relatively high-quality meshes but faces challenges with multi-object scenes and occasionally generates geometries inconsistent with input images, such limitation also observed in TripoSR without using any generative prior. Our method achieves better geometric fidelity and completeness through careful design of both the

Figure 4. **VAE Metrics with varying number of tokens.** We show the reconstructed mesh CD and IoU of our Pyramid VAE vs the original VAE in terms of different number of tokens.

Setting	F-Score ↑	CD ↓	NC↑
w/o Pyramid VAE	0.928	0.397	0.807
w/o condition aug	0.902	0.435	0.789
w/o MAR-HR	0.921	0.411	0.794
w/o rotation aug	0.934	0.369	0.821
full	0.944	0.351	0.835

Table 2. **Ablation study of different components in our method.** ↑ indicates higher is better, and ↓ indicates lower is better.

VAE architecture and generation pipeline.

4.4. Ablation Study

VAE Ablations We evaluate the VAE performance by analyzing latent token length and our multi-resolution pyramid design. The quantitative results in Fig. 4 demonstrate that reconstruction error decreases with increasing token length, though improvements beyond 1024 tokens become marginal. Our pyramid design consistently enhances reconstruction quality, with 1024 tokens under Pyramid VAE outperforming 2048 tokens without it. Based on this analysis, we select 256 tokens for MAR-LR and 1024 tokens for MAR-HR, striking an optimal balance between computational efficiency and generation quality. As shown in Fig. 6, more geometric details are shown in our Pyramid VAE (d) using less tokens compared with single-level VAE (c).

Generation Ablations Tab. 2 presents our ablation study for generation on several key components on GSO datset: Pyramid VAE, condition augmentation, MAR-HR, rotation augmentation. The results demonstrate that our Pyramid VAE enhances generation quality through its improved latent space, adding MAR-HR with condition augmentation significantly improves cascaded generation quality by mit-

Figure 5. **Ablation study on token resolution and model scaling strategies.** Results (a)-(h) demonstrate different model configurations and settings, with detailed analysis provided in the main text.

Figure 6. **Visual comparison of VAE reconstruction.** (a)-(c) show reconstruction results from single-level VAE compressed with 256, 1024, and 2048 latent tokens respectively. (d) demonstrates the result from our Pyramid VAE using 1024 tokens.

igating compounding error, while view augmentation reduces view ambiguity. We also evaluate our cascaded MAR design through comparison with a DiT implementation, as illustrated in Fig. 5. Our base model with 256 latent tokens (e) achieves superior geometry quality compared to the DiT version with the same latent tokens (a). When directly increasing to 1024 tokens, both our model (f) and DiT (b) show degraded performance due to convergence issues. While the cascaded model enhances detailed generation, error propagation from the low-resolution model introduces significant noise in both MAR (g) and DiT version (c). Our MAR-HR with condition augmentation (h) successfully upscales token resolution and achieves detailed generation, demonstrating clear advantages over the DiT version (d). This ablation study demonstrates that our cascaded MAR with condition augmentation offers an effective and efficient solution for scaling up the token resolution. Quantitative comparison are provided in the supplementary.

5. Conclusion

We present a new 3D generation paradigm that combines auto-regressive and diffusion models while addressing key challenges in scaling to longer tokens. Through a Pyramid VAE and cascaded training with condition augmenta-

tion strategy, we progressively refine low-resolution tokens into high-resolution ones. Both quantitative and qualitative results demonstrate the effectiveness of our method, highlighting the potential of auto-regressive 3D generation.

References

[1] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. In *CVPR*, 2022. 3, 4, 5

[2] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T. Freeman, Michael Rubinstein, Yuanzhen Li, and Dilip Krishnan. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 3

[3] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *ECCV*, 2024. 2

[4] Jinnan Chen, Chen Li, Jianfeng Zhang, Lingting Zhu, Buzhen Huang, Hanlin Chen, and Gim Hee Lee. Generalizable human gaussians from single-view image. *arXiv preprint arXiv:2406.06050*, 2024. 2

[5] Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Yanru Wang, Zhibin Wang, Chi Zhang, et al. Meshxl: Neural coordinate field for generative 3d foundation models. *arXiv preprint arXiv:2405.20853*, 2024. 3

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

[7] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv preprint arXiv:2204.11918*, 2022. 6

[8] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2020. 3

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021. 3, 4

[10] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NIPS*, 2020. 3

[12] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *arXiv preprint arXiv:2106.15282*, 2021. 5

[13] 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*, 2024. 2

[14] Tao Hu, Wenhang Ge, Yuyang Zhao, and Gim Hee Lee. X-ray: A sequential 3d representation for generation. *NeurIPS*, 2024. 2

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH*, 2024. 2

[16] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2

[17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023. 2

[18] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In *ICLR*, 2024. 2

[19] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024. 3, 4, 6

[20] Weiyu Li, Jiarui Liu, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024. 2, 6

[21] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2

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

[23] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 2

[24] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 3

[25] 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. 2

[26] Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *ICLR*, 2020. 3

[27] 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. 3

[28] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5

[29] William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022. 6

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 5

[31] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, 2024. 2, 3

[32] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023. 2

[33] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2

[34] 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. 3

[35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2022. 5

[36] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024. 3

[37] 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. 2, 6

[38] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024. 3

[39] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2, 6

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2023. 3

[41] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 2

[42] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024. 2

[43] 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. 2

[44] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, 2023. 6

[45] Desai Xie, Sai Bi, Zhixin Shu, Kai Zhang, Zexiang Xu, Yi Zhou, Sören Pirk, Arie Kaufman, Xin Sun, and Hao Tan. Lrm-zero: Training large reconstruction models with synthesized data. *arXiv preprint arXiv:2406.09371*, 2024. 2

[46] 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. 2, 6

[47] Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, and Kai Zhang. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023. 2

[48] Xu Yinghao, Shi Zifan, Yifan Wang, Chen Hansheng, Yang Ceyuan, Peng Sida, Shen Yujun, and Wetzstein Gordon. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621*, 2024. 2

[49] Zhengming Yu, Zhiyang Dou, Xiaoxiao Long, Cheng Lin, Zekun Li, Yuan Liu, Norman Müller, Taku Komura, Marc Habermann, Christian Theobalt, et al. Surf-d: High-quality surface generation for arbitrary topologies using diffusion models. *arXiv preprint arXiv:2311.17050*, 2023. 2

[50] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *arXiv preprint arXiv:2301.11445*, 2023. 2

[51] 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. *arXiv preprint arXiv:2406.13897*, 2024. 2, 3, 5

[52] 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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[53] Lingting Zhu, Jingrui Ye, Runze Zhang, Zeyu Hu, Yingda Yin, Lanjiong Li, Jinnan Chen, Shengju Qian, Xin Wang, Qingmin Liao, and Lequan Yu. Muma: 3d pbr texturing via multi-channel multi-view generation and agentic post-processing. *arXiv preprint arXiv:2503.18461*, 2025. 2