

# Unleashing Vecset Diffusion Model for Fast Shape Generation

Zeqiang Lai<sup>1,2\*</sup>, Yunfei Zhao<sup>2,3\*</sup>, Zibo Zhao<sup>2,4</sup>, Haolin Liu<sup>2</sup>, Fuyun Wang<sup>1</sup>  
 Huiwen Shi<sup>2</sup>, Xianghui Yang<sup>2</sup>, Qingxiang Lin<sup>2</sup>, Jingwei Huang<sup>2</sup>  
 Yuhong Liu<sup>2</sup>, Jie Jiang<sup>2</sup>, Chunchao Guo<sup>2†</sup>, Xiangyu Yue<sup>1†</sup>  
<sup>1</sup>MMLab, CUHK   <sup>2</sup>Tencent Hunyuan   <sup>3</sup>VISG, NJU   <sup>4</sup>ShanghaiTech  
<https://github.com/Tencent/FlashVDM>

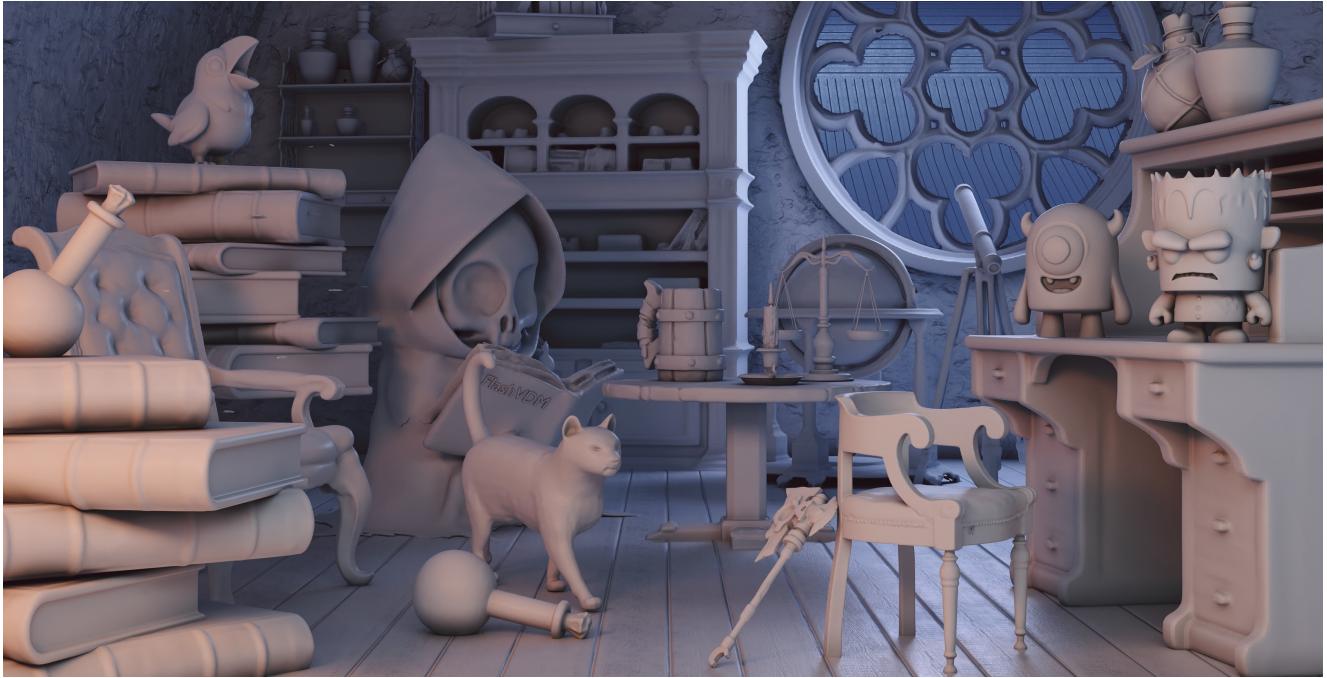


Figure 1. High-resolution 3D shapes generated by Flash Vecset Diffusion Model (FlashVDM) within 1 second.

## Abstract

3D shape generation has greatly flourished through the development of so-called “native” 3D diffusion, particularly through the Vecset Diffusion Model (VDM). While recent advancements have shown promising results in generating high-resolution 3D shapes, VDM still struggles at high-speed generation. Challenges exist because of not only difficulties in accelerating diffusion sampling but also VAE decoding in VDM – areas under-explored in previous works. To address these challenges, we present **FlashVDM**, a systematic framework for accelerating both VAE and DiT in VDM. For DiT, FlashVDM enables flexible diffusion sampling with as few as 5 inference steps and com-

parable quality, which is made possible by stabilizing consistency distillation with our newly introduced Progressive Flow Distillation. For VAE, we introduce a lightning vecset decoder equipped with Adaptive KV Selection, Hierarchical Volume Decoding, and Efficient Network Design. By exploiting the locality of vecset and the sparsity of shape surface in the volume, our decoder drastically lowers FLOPs, minimizing the overall decoding overhead. We apply FlashVDM to Hunyuan3D-2 [59] to obtain Hunyuan3D-2 Turbo. Through systematic evaluation, we show that our model significantly outperforms existing fast 3D generation methods, achieving comparable performance to the state-of-the-art while reducing inference time by over 45× for reconstruction and 32× for generation.

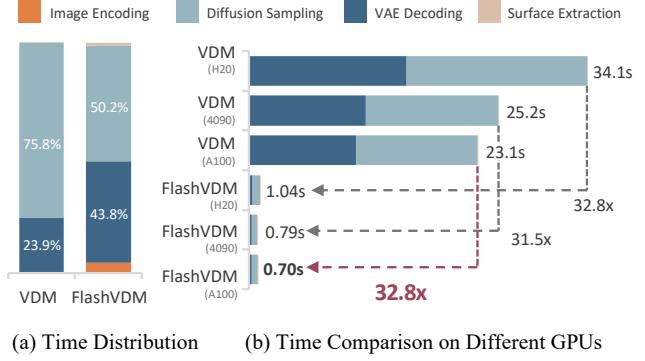
\* Equal contribution. † Corresponding authors.

## 1. Introduction

3D shape generation has greatly flourished through the development of so-called “native” 3D diffusion [46, 56], among which the Vecset Diffusion Model (VDM) [54, 58] receives prominent attention and applications due to its simplicity and scalability. While recent works [18, 56, 59] have demonstrated the capabilities of VDM in generating high-resolution and high-quality 3D shapes, VDM still struggles with high-speed generation, typically requiring over 30 seconds<sup>1</sup> per shape – far behind the development of image generation counterparts [26, 34, 35, 52]. The notable reasons behind the lags of VDM [54] against previous 2D predecessors [31] lies in not only the lack of research of diffusion distillation [26, 28] for few-step 3D generators, but also the acceleration of Variational Autoencoders (VAE).

Unlike the 2D VAE [6], which utilizes convolution for image compressing and decoding with structured latent, the VAE in VDM takes a fundamentally different approach. Its encoder compresses the point cloud of the mesh surface into a set of latent tokens—typically referred to as a *vec-set*—through cross-attention (CA), using a set of query tokens [54]. This approach, similar to Q-former [14] and Perceiver [12], offers strong compression capability but also introduces challenges in decoding. To overcome this, VDM employs a symmetric approach, constructing a set of volume queries that evaluate occupancy or Signed Distance Function (SDF) at each grid point using another CA. While the CA decoder design enables VDM to flexibly decode at arbitrary resolutions, the computational cost increases cubically with respect to resolution. For instance, at a typical resolution of 384, the number of query points exceeds 56 million—each requires a CA operation. Even worse, modern VDMs [18, 59] typically use a much larger set of latent tokens for key-value pairs to ensure high-resolution generation, which ends up making the decoding process even slower. As a result, VAE decoding in VDM demands a significant amount of inference time—even longer than diffusion sampling as shown in Fig. 2 (a).

On the other hand, despite the significant advancements in fast image generation through diffusion distillation [26, 28, 34, 35, 52], pushing the speed to real-time, research into native 3D diffusion distillation remains scarce. Most existing distillation methods [26, 34, 52] were originally designed for images, with some adaptations made for video [53, 62]. Many techniques, such as LPIPS loss in Consistency Distillation (CD) [23, 38, 40] and GAN designs [43, 52], are specifically tailored for images. However, the representation of 3D meshes and images differs significantly, making it challenging to adapt these techniques. Additionally, the vastly different latent spaces in VDMs against image DMs [31] could drastically alter the



(a) Time Distribution      (b) Time Comparison on Different GPUs

Figure 2. (a) VDM exhibits different distribution with image DM with much larger percentage in VAE decoding. (b) FlashVDM enables fast shape generation within 1 second on a consumer GPU.

training dynamics of the distillation, which makes previous strategies unsuitable, potentially leading to instability and unsatisfactory results.

To address the aforementioned problems, we present *FlashVDM*, a framework for transforming pretrained VDM into a high-fidelity and high-speed generator with over **45x speedup** in VAE decoding and **32x speedup** overall. Our framework consists of two main components: 1) diffusion acceleration and 2) VAE acceleration.

For diffusion acceleration, we carefully analyzed the problems in direct application of CD [40, 43] for VDM and found that the core issue lies in the instability of the target network, which degrades the training and causes unsatisfactory results. As a remedy, we propose *Progressive Flow Distillation*, a multi-stage distillation method for VDM. It first decouples the CD distillation by warming up with guidance distillation. This prevents the target bias and fluctuations from misleading the student. Then, we carefully ablate the design choices including loss and training strategies in CD, leading to even more stable step distillation. Finally, we introduce adversarial finetuning, leveraging real 3D data as supervision, to help distilled models produce smoother and more accurate results within limited steps. Together, our distilled model could achieve comparable results with only 5 Number of Function Evaluation (NFE).

For VAE acceleration, we introduce three techniques, including two training-free methods—*Adaptive KV Selection* and *Hierarchical Volume Decoding*—to reduce both the number of number of key-value pairs and the queries. These techniques effectively exploit the locality of VDM point-queries and the sparsity of shape surface in the volume space, resulting in a highly efficient CA with a 97.1% FLOPs reduction in total. Additionally, we introduce an *Efficient Decoder Architecture*. By fixing the encoder and fine-tuning only the decoder, our new design further reduces FLOPs by 76.6% for each query. With these three improvements combined, our final VAE decoder achieves over a

<sup>1</sup>Measured by the default setting of Hunyuan3D-2 [59].

45× speedup, significantly reducing the decoding time from 22.3s to just 0.49s.

In summary, our contributions are listed as follows:

- We introduce FlashVDM, a framework that converts pre-trained VDMs into high-speed and -fidelity generators.
- We present progressive flow distillation for VDM, significantly improving the stability and quality of the distilled model, achieving comparable results with only 5 NFE.
- We propose a novel algorithm, hierarchical volume decoding with adaptive KV selection and an efficient decoder architecture, obtaining 45× speedup in decoding.
- We apply FlashVDM to Hunyuan3D-2 [59] and obtain Hunyuan3D-2 Turbo, a high-resolution shape generator that matches its teacher’s shape quality with over 32× speedup, achieving 1 second per shape.

## 2. Related Works

**Diffusion Acceleration.** Step distillation [8, 25, 28, 33, 34, 40, 47, 50, 52, 61] is a popular technique for accelerating diffusion models, with early works like Progressive Distillation [28, 33] training student models by halving sampling steps step-by-step. More recent approaches, such as consistency models [39, 40, 43] and score distillation [34, 52], enforce self-consistency or match distributions between student and teacher models. GAN-based approaches [34, 44] further refine score distillation by augmenting loss functions with adversarial objectives. Several works, including UFOGen [48] and LADD [35], combine GANs with pretrained diffusion models as discriminators. Other research focuses on reducing redundancy in sampling processes, such as DPM-Solver [24], SnapFusion [17], and DeepCache [27]. In this work, we introduce progressive flow distillation to address practical issues in 3D foundation models, an area not yet explored in previous research.

**VAE Acceleration.** The proposed hierarchical volume decoding is generally related to octree decoding [32]. However, we noted that vanilla octree decoding introduces artifacts and holes, whereas our method is nearly lossless. Our adaptive KV selection is broadly related to previous work on token merging [1, 2] in DiT inference, though our approach is specifically designed for VAE acceleration in VDM without the merge operation. To the best of our knowledge, we are the first to explore the efficient design of a VDM decoder. DC-AE [4] is one of the works for image diffusion models [4], but it focuses on achieving a higher compression ratio, which is different from our objective.

**Fast 3D Generation.** Previous works on fast 3D generation are based on feedforward 3D reconstruction methods [9], which require only a single network evaluation. Benefiting from this property, TripoSR [42] and SF3D [3] can generate a mesh from a single image in under 1 second, but the resulting mesh quality is limited. To address this, multiview images are often used [15, 41], but this re-

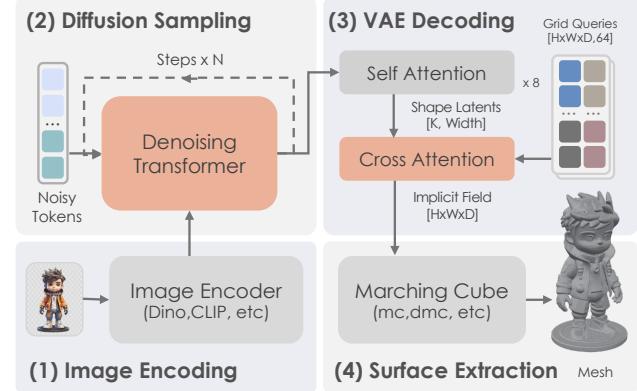


Figure 3. Illustration of four main stages of VDM.

quires a Multi-View Diffusion (MVD) model [21, 36, 37]. In response, SPAR3D [11] replaces MVD with a small point cloud generator, while Turbo3D [10] distills MVD into a four-step generator. Nevertheless, all these feedforward methods are still limited in mesh quality compared to diffusion-based 3D generators [56, 59].

## 3. Method

In this section, we first provide an introduction to VDM pipeline and inference time distribution. Then, we illustrate our approaches for accelerating VAE decoding in Sec. 3.2 and diffusion sampling in Sec. 3.3, respectively.

### 3.1. Preliminary of VDM

**Pipeline Overview.** Here, we provide a brief introduction to the inference pipeline of VDM. We refer interested readers to [5, 54, 56, 59] for more details. As shown in Fig. 3, the inference of VDM consists of four stages: 1) image encoding: an image encoder [29, 30] is used to extract conditional features; 2) diffusion sampling: iteratively calling a denoising transformer to predict shape latent based on conditional features. 3) volume decoding: the predicted shape latent is decoded into volume SDF via several layers of self-attention and a cross-attention. 4) surface extraction: marching cube algorithm [22] is used to extract polygon mesh from the decoded volume.

**Inference Time Distribution.** The distribution of inference times is shown in Fig. 2(a). We could observe that diffusion sampling (23.9%) and volume decoding (75.8%) account for the majority (99.7%) of the inference time for VDM. However, image encoding and surface extraction remain significant when the total inference time is reduced to less than one second. As a first step, we implement several engineering acceleration techniques, including FP8 attention with SageAttention2 [55], static graph optimization with `torch.compile`, and *etc.*, which decrease the processing time to 25.09s per shape, as shown in Fig. 5.

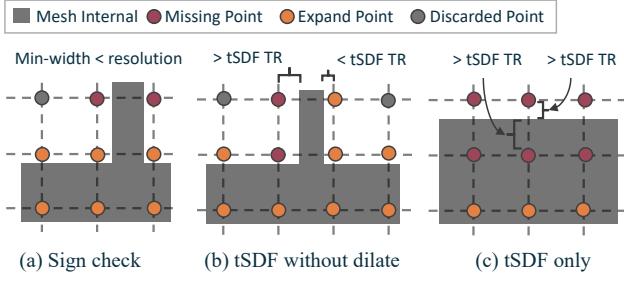


Figure 4. The corner cases in hierarchical volume decoding.

### 3.2. Lightning Vecset Decoder

The majority (99.9%) of the computational cost in the original VDM decoder is concentrated in its final cross-attention layer, which is evaluated tens of millions of times. To optimize this layer, we propose three key techniques, each addressing a different aspect: 1) hierarchical volume decoding to reduce the number of queries, 2) adaptive KV selection to minimize the number of keys and values, and 3) an efficient decoder design to reduce computational overhead in each cross-attention operation.

**Hierarchical Volume Decoding.** To decode shape latents into a mesh, the existing VDM decoder relies on dense volume decoding and marching cube algorithms [22], which has cubic complexity with respect to the resolution. It raises challenges for balancing speed and mesh quality. However, unlike image decoders [6] that must predict RGB values for every pixel, the VDM decoder only needs to determine high-resolution SDF values near the shape surface – all voxels far from the surface can be simply classified as inside or outside. Inspired by this characteristic, we propose hierarchical volume decoding that only increases resolution near the surface, thereby reducing the queries by 91.4%.

Specifically, instead of directly decoding the SDF volume at the target resolution (*e.g.*, 384), we begin by decoding at a smaller resolution (*e.g.*, 75), which provides a coarse SDF volume. Using this coarse volume, we can identify which voxels intersect the shape surface based on the following criterion:

$$f(x, y, z) = \begin{cases} 1 & \text{if } \exists(i, j, k) \in N(x, y, z) \\ & \text{such that } f(i, j, k) \neq f(x, y, z), \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where  $N(x, y, z)$  represents the set of adjacent voxels to  $(x, y, z)$  and  $f(x, y, z) = 1$  indicates the voxel intersects with the surface. In short, a voxel does not intersect with the surface only if all its adjacent voxels are either inside or outside. For voxels that do intersect the surface, we subdivide them into higher resolutions and compute the SDF values through cross-attention once more. This process is

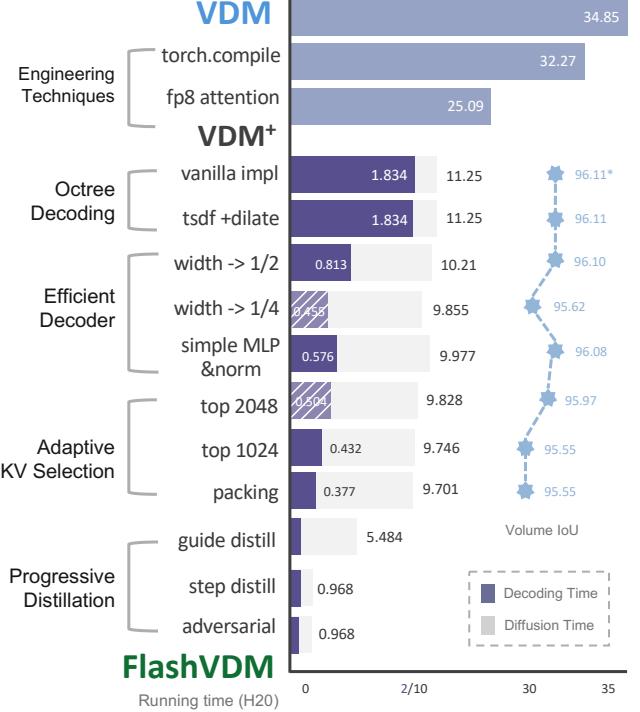


Figure 5. Step-by-step case study of FlashVDM acceleration techniques and their effects on inference time and IoU.

repeated until we reach the target resolution. Since the surface shape is highly sparse within the volume, the number of effective queries is significantly reduced.

Naively, previous steps could be considered a type of octree decoding [32]. However, we find that it produces artifacts and holes when applied to VDM in practice. To prevent degradation, we need to consider some corner cases. First, when the resolution is low, for very thin meshes, two adjacent voxels may lie on opposite sides of the mesh surface, both having the same sign, as shown in Fig. 4 (a). This causes the missing voxels for resolution increment, leading to holes in the surface. To address this, we use tSDF as supervision during VAE training to help determine whether a voxel is near the mesh surface. Then, during hierarchical decoding, we append all voxels under a certain tSDF threshold (TR). With this strategy, the lowest resolution in hierarchical decoding can be reduced to  $w + u * 2$  where  $w$  is the minimal mesh width and  $u$  is the tSDF TR. Second, failure cases exist even with tSDF augmentation as shown in Fig. 4 (b,c), *e.g.*, uneven distribution of surface between two adjacent voxels and *etc.* Therefore, we perform a dilation operation after identifying the intersected voxels to prevent unexpected missing points. We leave the choice of tSDF value and more details to Appendix B.

**Adaptive KV Selection.** For each position in volume decoding, a cross-attention operation is performed between

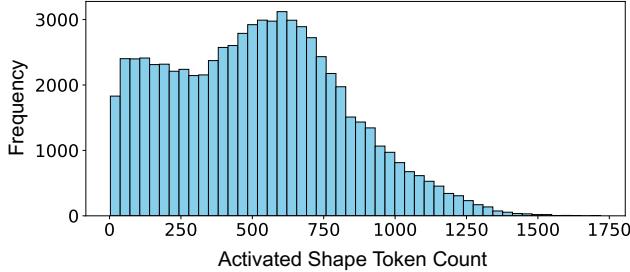


Figure 6. Histogram of activated token counts with non-zero attention of 70,800 regions from 250 cases.

the queries and key-value pairs (computed from shape latents). Previously, we significantly reduced the number of queries. To further minimize computational overhead, we introduce adaptive KV selection, which reduces the number of key-value pairs. Our method leverages the observation that the correlation between spatial queries and shape latents exhibits strong locality – adjacent spatial points tend to attend to similar latent tokens, with the attention concentrated on a small subset of tokens (denoted as activated tokens). To illustrate, we show two histograms in Figs. 6 and 7. As seen, most regions in 3D volumes attend to at most 1,000 tokens – one-third of the original 3,072, and different regions focus on different set of activated tokens. Moreover, our statistic reveals that the average activated token count for a single query is around only 10, which also indicates strong locality in shape latents.

The proposed adaptive KV selection is training-free and able to be combined with hierarchical volume decoding. For simplicity, we first show the algorithm with vanilla volume decoding. In this case, we first divide the entire volume into smaller sub-volumes. Then, within each sub-volume, we uniformly sample a small set of queries to compute their attention scores relative to the keys. To select most correlated key-value pairs, we could simply average the attention score of all queries and select the TopK, or we could merge the TopN of each query to obtain TopM, where N is a small number (*e.g.*, 50) that can include most activated tokens with non-zero attention. The latter one is slightly better but also slightly slower (5%). For other queries within the same sub-volume, we only use the Top-K/M key-value pairs for attention computation. Since the attention scores are estimated using a small set of queries and the number of activated shape latents are much smaller, the overall computation can be reduced by 34%, as shown in Fig. 5.

The combination with hierarchical volume decoding needs some extra consideration to ensure performance. The core obstacle lies in the requirement that subvolume should be small to maintain locality. Within hierarchical decoding, the effective queries could be even fewer in a subvolume. Thereafter, naive implementation that process each subvol-

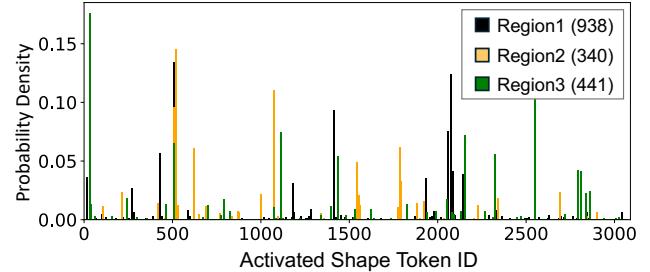


Figure 7. Normalized histogram of activated shape tokens with non-zero attention at different regions. Zoom in for a better view. The numbers in the legend indicate the number of activated tokens.

ume one-by-one would not maximize GPU utilization. As a remedy, we design a packing operation, which precompute the target queries in each volume, pack queries of different subvolume, and process them together. More details on the analysis and the implementation lives in Appendix C.2.

**Efficient Decoder Design.** Once we have reduced the QKV in CA, further acceleration of the attention operation itself becomes challenging algorithmically. Thereby, our next approach focuses on optimizing the decoder network design. Most existing vectorset decoders [16, 54, 58] share nearly identical architectures, typically consisting of several self-attention (SA) layers and a cross-attention layer, as shown in Fig. 3, with SA layers evaluated only once. Therefore, our goal is to simplify the final CA layer as much as possible, which we believe could be replaced with a simpler one as it only needs to classify the inside/outside status of each position. To this end, we reduce the network width, decrease the MLP expansion ratio, and remove several LayerNorm layers. Through careful ablation studies, we found that these reductions have minimal impact on reconstruction quality while boosting speed by 3.2 $\times$ .

### 3.3. Progressive Flow Distillation

Similar to image and video generation [13, 19, 31], we can reduce the sampling steps of VDM through distillation [26, 62]. Initially, we experimented with PCM [43], but we found it hard to yield satisfactory results on VDM. To address this, we propose progressive flow distillation as shown in Fig. 8, systematically identifying and refining key design choices for effective distillation.

**Consistency Flow Distillation (CFD).** Consistency Distillation (CD) [40] is a well-known distillation method for reducing the diffusion sampling steps, though its application to 3D diffusion has yet to be explored. The core of CD is to enforce a consistency property – any point along the ODE trajectory maps to the same target point. To achieve this, CD uses a time difference method, where a starting point  $\mathbf{x}_{t_n}$  is first selected for the target prediction of the student model  $\mathbf{x}_0^{t_n} = \mathbf{f}_\theta(\mathbf{x}_{t_n}, t_n)$ . For flow-based mod-

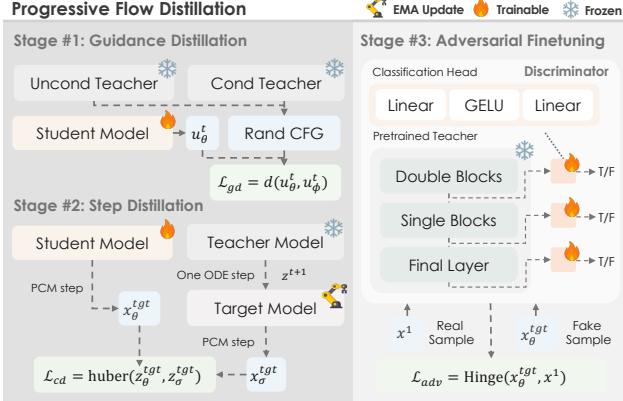


Figure 8. Training pipeline for Progressive Flow Distillation.

els, e.g., Hunyuan3D-2 [59], the Euler solver could be employed with a large discretization step  $t_n - t_0$  to compute the target  $\mathbf{x}_0^{t_n}$ . For GT target, the teacher model predicts the next point  $\hat{\mathbf{x}}_{t_{n+1}}^\phi$  by running one discretization step of an ODE solver, after which the student model predicts the GT target at next timestep  $x_0^{t_{n+1}} = f_\theta(\hat{\mathbf{x}}_{t_{n+1}}^\phi, t_{n+1})$ . The consistency property is enforced by the following loss,

$$\mathcal{L}_{cf}( \theta ) := \mathbb{E} \left[ d \left( f_\theta(\mathbf{x}_{t_n}, t_n), f_{\theta^-}(\hat{\mathbf{x}}_{t_{n+1}}^\phi, t_{n+1}) \right) \right], \quad (2)$$

with a boundary constraint  $f(\mathbf{x}_\epsilon, \epsilon) = \mathbf{x}_\epsilon$ , where  $d$  is a distance function and  $\theta^-$  is a regular copy of student model named as target model.

**Stabilizing Target Model.** In practice, directly applying the distillation method described above does not yield satisfactory results, as shown in Fig. 12 (w/o GD distill). Our analysis reveals that the main challenge lies in the stability of the target model. During training, the student model continually tries to mimic the output of the target model, making it highly sensitive to changes in the target. As a result, we observe significant fluctuations in the predictions throughout the training process. To stabilize the target model, we introduce a progressive training strategy, in which 1) we first perform guidance distillation and initialize the student model as a distilled version. This approach differs from distillation in image models, where guidance distillation can be applied simultaneously with step distillation without issues. We hypothesize that this gap arises from the differences between 2D and 3D models and the more complex optimization landscape in 3D models. 2) Besides, we find that the EMA update of the target model is crucial for stabilizing VDM, which contrasts with a recent study on 2D generation as well [38]. 3) Moreover, we replace commonly used L2 loss with Huber loss as it is less sensitive to outliers, which makes training more stable. 4) Finally, we introduce a multi-stage-multi-phase consistency distillation strategy based on PCM [43], in which a single-phase

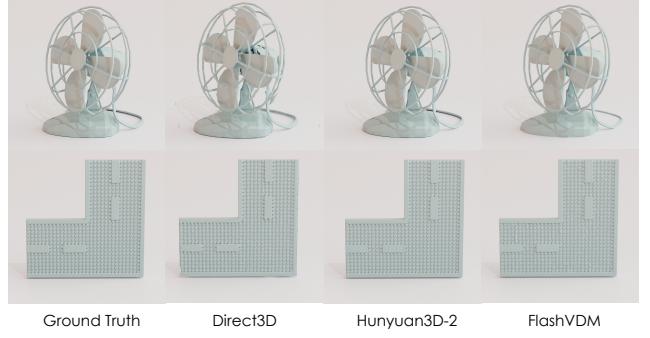


Figure 9. Visual comparison of shape reconstruction methods.

	V-IoU( $\uparrow$ )	S-IoU( $\uparrow$ )	Time(s, $\downarrow$ )
3DShape2VecSet [54]	87.88%	84.93%	16.43
Michelangelo [58]	84.93%	76.27%	16.43
Direct3D [45]	88.43%	81.55%	3.201
Hunyuan3D-2 [59]-1024	93.60%	89.16%	16.43
└ with FlashVDM	91.90%	88.02%	0.382
Hunyuan3D-2 [59]-3072	96.11%	93.27%	22.33
└ with FlashVDM	95.55%	93.10%	0.491

Table 1. Numerical comparisons of shape reconstruction methods.

finetuning is performed after five phases pretraining.

**Aligning Real Data with GAN.** With the proposed improvements, our CFD is able to reduce the sampling steps to just 5 while still producing decent results. However, we found that in some cases, the mesh quality is difficult to match the teacher’s outputs, which might be attributed to the self-distillation nature of the consistency model. To address this, we seek to further enhance performance by incorporating supervision from ground-truth 3D data through adversarial training [7]. Specifically, we initialize the generator using the model distilled in the previous stage. Inspired by previous works [35, 48], our discriminator operates directly in latent space, eliminating the need for an expensive decoding process, and leverages the intermediate features of the pretrained diffusion model. As shown in Fig. 8, we first extract token sequences from specified attention layers, and then apply independent discriminator heads. Unlike prior works [35, 43, 48, 51] that use noised latents, we did not observe significant improvements with noise, so we opt to use ground-truth latents for simplicity. The discriminator is trained with a hinge adversarial loss [20] as,

$$\mathcal{L}_{adv}(\theta, \gamma) = \text{ReLU}(1 + \mathcal{D}_\gamma(\mathbf{x}_0)) + \text{ReLU}(1 - \mathcal{D}_\gamma(\mathbf{x}_0^{t_n})), \quad (3)$$

which distinguishes between ground truth latent  $\mathbf{x}_0$  and latent produced by our generator  $\mathbf{x}_0^{t_n} = f_\theta(z_{t_n}, t_n)$ . Following [43, 51], the final objective is a combination of the adversarial objective and consistency loss  $\mathcal{L} = \mathcal{L}_{cf} + \lambda \mathcal{L}_{adv}$  where  $\lambda$  is set to 0.1.

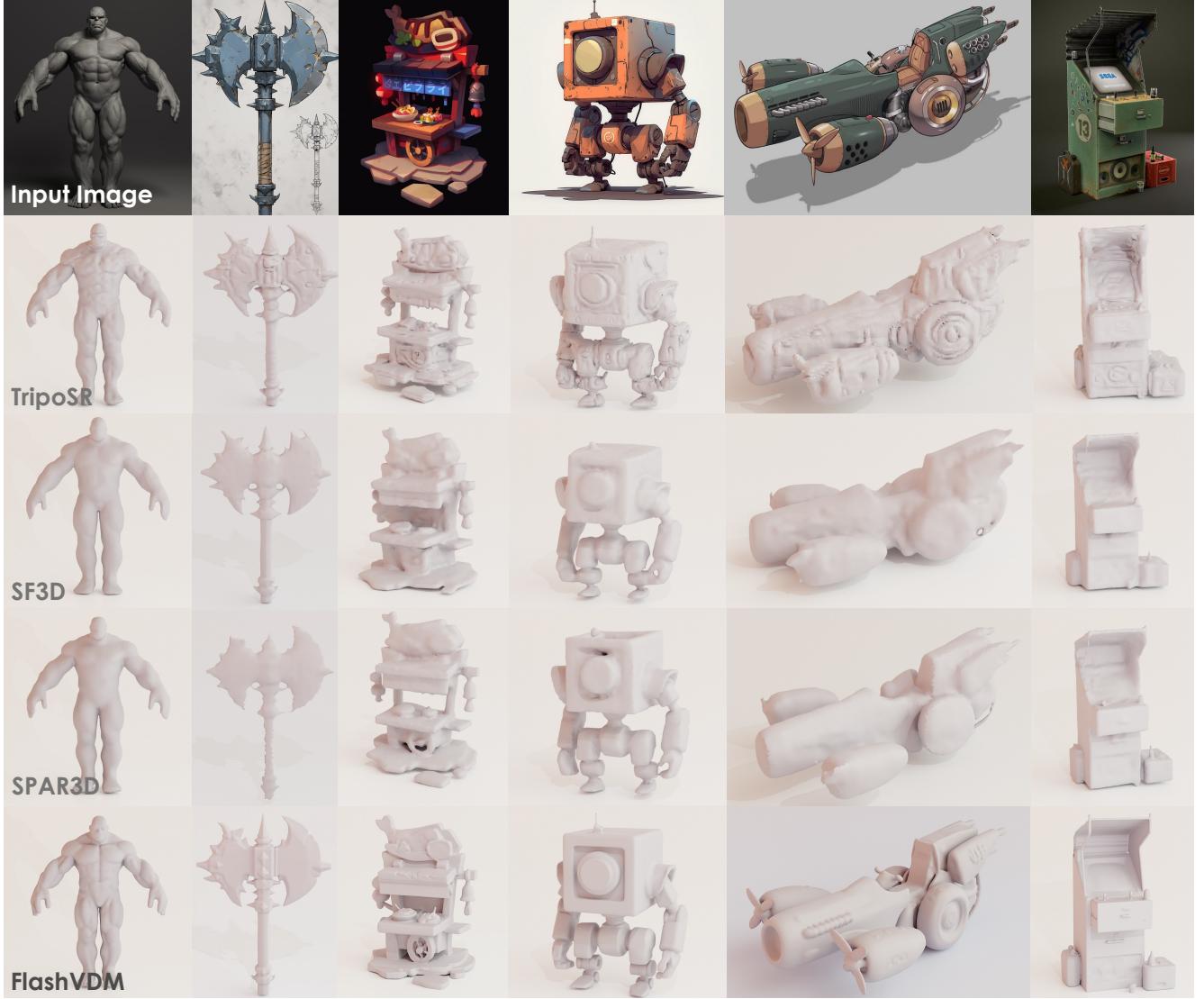


Figure 10. Visual comparison of image-to-3D generation between the proposed FlashVDM and other fast 3D generation methods.

## 4. Experiments

In this section, we apply the proposed FlashVDM to Hunyuan3D-2 [59], which is currently a state-of-the-art open-source VDM. We evaluate our approach in terms of both VAE reconstruction and diffusion generation. We also provide ablation studies of the proposed techniques.

### 4.1. Reconstruction

**Metrics.** We utilize the volume and surface IoU metric to assess the impact of our acceleration techniques on VAE reconstruction performance. The running time is measured at the resolution of 380.

**Comparison.** We compare our method with three competing method, *i.e.*, 3DShape2VecSet [54], Michelan-

gelo [58], and Direct3D [45]. We evaluate two resolutions, *i.e.*, 1,024 and 3,072, for our fast version and the base model Hunyuan3D-2 [59]. The numerical comparison is shown in Tab. 1. It demonstrates that our method outperforms all competing methods and preserve the quality of base model with less than 1% IoU drop while obtaining over 45x speedup. Fig. 9 shows the visual comparison, which also indicates minimal degradation in quality.

### 4.2. Generation

**Metrics.** To evaluate shape generation performance, we adopt ULIP-I [49] and Uni3D-I [60] to compute the similarity between the generated mesh and input images.

**Comparison.** Three latest fast 3D generation methods, *i.e.*, TripoSR [42], SF3D [3], and SPAR3D [11], and two

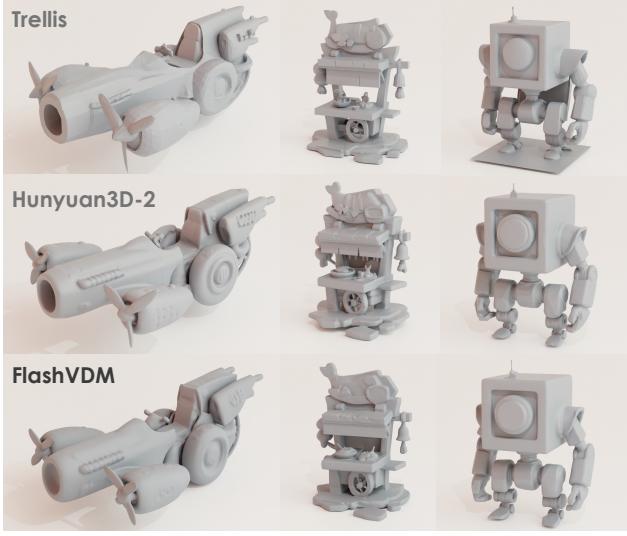


Figure 11. Visual comparison of image-to-3D generation between the proposed FlashVDM and other 3D diffusion methods.



Figure 12. Ablation study of our progressive flow distillation.

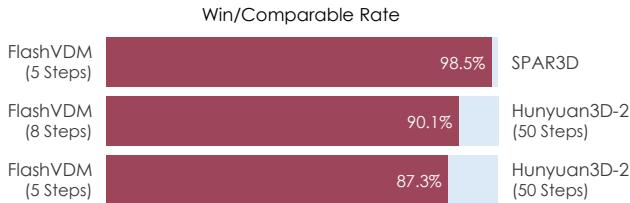


Figure 13. User study of FlashVDM against different methods.

state-of-the-art methods Hunyuan3D-2 [59] and Trellis [46] are selected for comparison. The numerical comparison is shown in Tab. 2 and the visual comparison is shown in

	<b>ULIP-I(↑)</b>	<b>Uni3D-I(↑)</b>	<b>Time(s,↓)</b>
TripoSR [42]	0.0642	0.1425	0.958
SF3D [3]	0.1156	0.2676	0.212
SPAR3D [11]	0.1149	0.2679	1.296
Trellis [46]	0.1267	0.3116	7.334
Hunyuan3D-2 [59]	0.1303	0.3151	34.85
└ with FlashVDM	0.1260	0.3095	1.041

Table 2. Numerical comparisons of shape generation methods.

	<b>V-IoU(↑)</b>	<b>S-IoU(↑)</b>	<b>Time(s,↓)</b>
VAE Baseline	96.11%	93.27%	22.33
+ Hierarchical Decoding	96.11%	93.27%	2.322
+ Efficient Decoder	96.08%	93.13%	0.731
+ Adaptive KV Selection	95.55%	93.10%	0.491

Table 3. Step-by-step ablation of our lightning vecset decoder.

Figs. 10 and 11. It can be observed that our method retain most of capability of the base model, while achieves the best results with a large margin against other fast methods.

**User Study.** We include results of three user studies in Fig. 13. For comparison between FlashVDM and SPAR3D [11], we ask participants to determine which one is better. For comparison between FlashVDM and Hunyuan3D-2 [59], we ask whether there is a huge difference. As seen, FlashVDM is preferred over SPAR3D [11] in almost all testcases, and FlashVDM is comparable to its base model Hunyuan3D-2 [59] at 5 steps and the preference rate increase with more steps.

### 4.3. Ablation Study

**Effectiveness of Lightning Vecset Decoder.** The results of step-by-step ablation is shown in Tab. 3, in which we add each component incrementally. It can be observed that hierarchical decoding provides a 10x speedup with no degradation in quality. The efficient decoder achieves an extra 3x speedup with almost no degradation, while adaptive KV selection results in a 30% speedup with minimal degradation.

**Effectiveness of Progressive Flow Distillation.** We present an ablation study to demonstrate the effectiveness of progressive training strategies. The visual comparison is shown in Fig. 12. As seen, the original VDM completely fails at 5 steps, while FlashVDM achieves results comparable to the original 50-step VDM. Additionally, we observe that distillation fails without guidance distillation as a warm-up step, and performance degrades without the use of EMA. Finally, adversarial fine-tuning is shown to help generate smoother and more accurate shapes. More detailed ablation can be found in the Appendix D.

## 5. Conclusion

In this work, we introduce *FlashVDM*, a general framework for accelerating a pretrained VDM [18, 59]. Our framework encompasses not only a progressive flow distillation method for distilling VDM into a few-step generator, but also several training-free inference techniques and an efficient, lightweight vecset decoder that significantly reduces the FLOPs of decoding. We apply our framework to the state-of-the-art image-to-3D VDM, Hunyuan3D-2 [59], obtaining Hunyuan3D-2 Turbo. Our evaluation demonstrates that FlashVDM excels in both reconstruction and generation, achieving 45 $\times$  and 32 $\times$  speedups, respectively. To the best of our knowledge, FlashVDM is the first work to push large-scale shape generation into the millisecond range, which opens up new possibilities for interactive applications of 3D generative models.

## A. Implementation Details

**Decoder Finetuning.** The efficient vecset decoder illustrated in Sec. 3.2 is fine-tuned by freezing the vecset encoder. In [59], the decoder is made up of 8 self-attention layers and 1 cross-attention layer. Since our design only alters the cross-attention layer, we initialize self-attention layers as before. Both self- and cross-attention layers are trained during the finetuning. We use a constant learning rate of  $1 \times 10^{-4}$  and a batch size of 256. The decoder could quickly converge to a pretty good one with 300k steps, but we find longer training to 800k steps converges better, leading to nearly identical performance to the original one.

**Diffusion Distillation.** The batch size is always 256 for different stages in progressive flow distillation. Following [26, 28], the guidance distilled model is conditioned on the guidance strength  $w$ , which is injected into the diffusion backbone with a similar approach as timestep. During training,  $w$  is randomly select from  $w \sim U[2, 8]$ . The learning rate is set to  $1 \times 10^{-6}$ . The model is trained with 20k steps. For step distillation, we set  $\lambda$  of huber loss to  $1 \times 10^{-3}$ , and the guidance strength is set to a constant of 5.0. Following [26], we also use the skipping-step technique with  $k = 10$ . We utilize multiphase [43] techniques to train the model for 20k steps with 5 phases and a learning rate of  $1 \times 10^{-6}$  and then finetune the model for 8k steps with a learning rate of  $1 \times 10^{-7}$ . The EMA decay rate is set to 0.999. For adversarial fine-tuning, we keep the distillation loss of the previous stage and set the adversarial loss weight to 0.1. The learning rate is set to  $1 \times 10^{-7}$  for generator and  $1 \times 10^{-6}$  for discriminator. We train 5k steps for this stage.

## B. Details of Hierarchical Volume Decoding

**Effect of Dilate and tSDF.** Fig. 14 compares the reconstruction with and without dilate and tSDF strategy. It can be seen that dilate+tSDF is mandatory for reconstructing complete mesh without holes.

**Implementation and Practical Consideration.** The overall pseudocode for hierarchical volume decoding is shown in Algorithm 1. In practice, we set the tSDF threshold  $\eta = 0.95$  and the isosurface threshold  $\gamma = 0.0$ . The dilate operation is implemented using a 3D convolution with a kernel size of 3. At the final resolution, the total number of points increases significantly, thus the `FindNear` operation would introduce many redundant points. To address this, we omit the `FindNear` operation while `Expand` twice, striking a balance between speed and quality. Practically, we find this strategy has minimal impact on the overall quality while speeding up slightly.

---

### Algorithm 1 Hierarchical Volume Decoding.

---

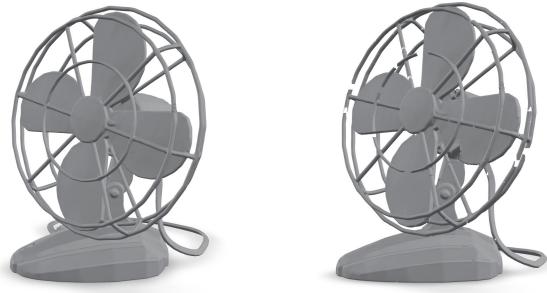
**Input:** An implicit function  $f(\mathbf{p})$  that evaluates the SDF at position  $\mathbf{p}$ . Target resolution  $r$ . Shape latents  $Z$ , tSDF threshold  $\eta$ , isosurface threshold  $\gamma$ .

**Output:** A signed distance field (SDF)  $S \in \mathcal{R}^{r \times r \times r}$ .

```

1:  $R = \text{GetResolutions}(r)$   $\triangleright$  List of cascade resolution.
2:  $P_{R[0]} = \text{GenGridPoints}(R[0])$ 
3:  $S_{R[0]} = \text{QueryField}(P_{R[0]}, Z)$ 
4: for  $i = 1$  to  $\text{len}(R)$  do
5:    $\hat{P}_{R[i]} = \text{FindIntersect}(S_{R[i-1]}, \gamma)$ 
6:    $\hat{P}_{R[i]} += \text{FindNear}(S_{R[i-1]}, \eta)$ 
7:    $\hat{P}_{R[i]} = \text{Dilate}(\hat{P}_{R[i]})$ 
8:    $P_{R[i]} = \text{Expand}(\hat{P}_{R[i]})$ 
9:    $S_{R[i]} = \text{QueryField}(P_{R[i]}, Z)$ 
10: end for
11: return  $S_{R[-1]}$ 
```

---



(a) Octree Volume Decoding      (b) Octree Volume Decoding  
w/o Dilate+tSDF

Figure 14. Comparison of reconstruction results with and without dilate and tSDF strategy for hierarchical volume decoding.

## C. Details of Adative KV Selection

### C.1. Analysis of Locality.

**Activated Tokens Across Different Cases in the Same Region.** Fig. 15 presents a histogram of the activated shape token count across 300 different test cases. As observed, different regions and cases activate different sets of tokens. This suggests that locality is case-dependent rather than region-dependent. In other words, the same region in different cases does not consistently share a similar set of activated tokens.

### Distribution of Activated Tokens Within a Case.

Fig. 16 shows the histogram of the total number of activated tokens within a case, based on 200 cases. It can be observed that most cases contain over 3000 tokens, with a maximum of 3072 tokens. This further confirms that the phenomenon of having fewer activated tokens per region is due to token locality, rather than token redundancy.

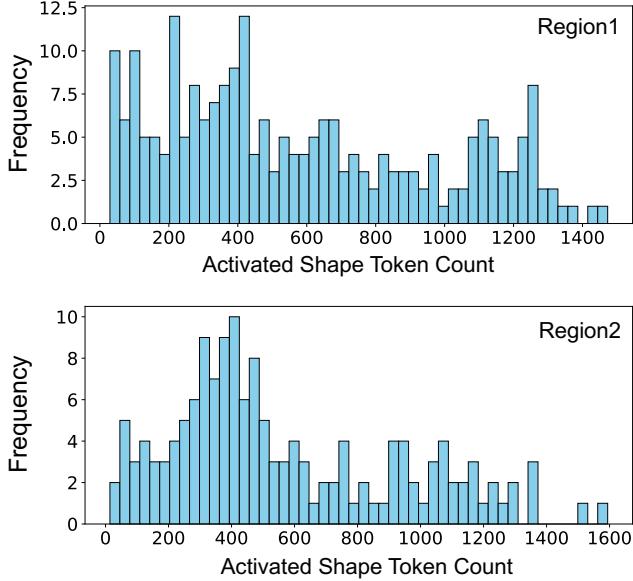


Figure 15. Histogram of activated token counts within different regions, measured with 300 cases.

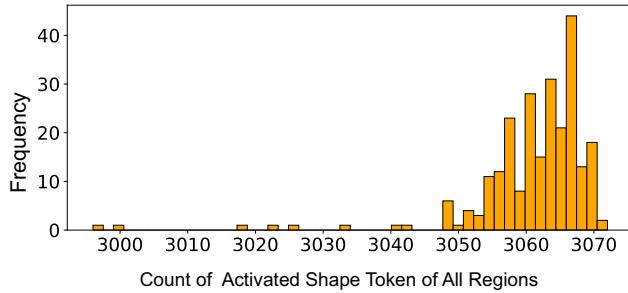


Figure 16. Histogram of the number of total activated token within all regions, measured with 200 cases.

**IoU Changes with Respect to TopK Tokens.** Fig. 17 shows the relationship between volume IoU and the number of TopK tokens, with all methods utilizing hierarchical volume decoding. The results for Original and FlashVDM differ due to the use of the efficient decoder. It can be observed that FlashVDM(r4) closely matches the curve of Original(r4), suggesting that our efficient decoder design preserves most of the reconstruction ability. Additionally, we notice that r16 performs significantly better than r4, highlighting the strong locality of attention between queries and shape tokens. Higher resolution corresponds to smaller subvolumes, resulting in improved locality. Interestingly, r16 maintains a similar IoU even with just 16% (512/3072) of the tokens.

## C.2. Implementation

**Combination with Hierarchical Volume Decoding.** In Sec. 3.2, we briefly introduce the combination of Adaptive

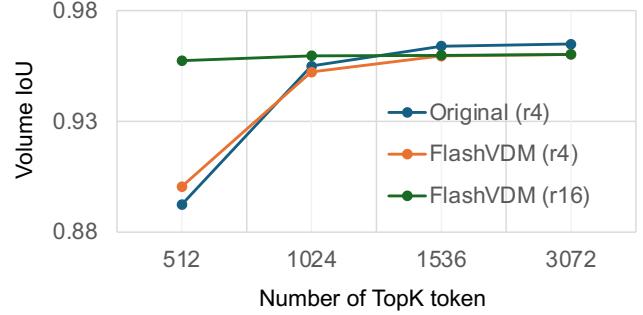


Figure 17. The graph shows the relationship between volume IoU and the number of TopK tokens. r4 denotes the volume is divided into  $4^3$  subvolumes, and r16 denotes  $16^3$  subvolumes.

---

### Algorithm 2 Adaptive KV Selection.

---

**Input:** Query  $Q \in \mathcal{R}^{N \times D}$ , Key  $K \in \mathcal{R}^{M \times D}$ , and Value  $V \in \mathcal{R}^{M \times D}$  of cross attention, the number of queries  $n \ll N$  for estimating TopK correlated KV.

**Output:** Attention result  $O \in \mathcal{R}^{N \times D}$ .

- 1:  $\hat{Q} = \text{Sample}(Q), \hat{Q} \in \mathcal{R}^{n \times D}$
- 2:  $M = \text{Mean}(\hat{Q} \times \hat{K}^T), M \in \mathcal{R}^{n \times M}$
- 3:  $\hat{K}, \hat{V} = \text{TopK}(M, K, V), \hat{K}, \hat{V} \in \mathcal{R}^{k \times D}$
- 4:  $O = \text{Attention}(Q, \hat{K}, \hat{V})$
- 5: **return**  $O$

---

KV Selection (AKVS) and hierarchical volume decoding. Here, we provide a more detailed explanation of the implementation and background. AKVS can be naively implemented as shown in Algorithm 2. The algorithm consists of four main steps: sampling queries, computing the mean attention score, selecting Top-K, and performing attention. Since the attention score is computed from the sampled queries, we need to feed the queries subvolume by subvolume, with queries being spatially close to one another, to keep locality.

For the original volume decoding, this can be easily achieved by changing the chunk-splitting method to a subvolume-splitting method, as the original method also uses chunk-splitting to reduce memory requirements. However, to maintain locality, the chunk size must be much smaller, which may be too small for efficient GPU acceleration. Additionally, with hierarchical volume decoding, the number of queries in each subvolume can be even smaller. To address this, we propose to pre-divide the subvolume and concatenate all queries of each subvolume. Instead of processing each subvolume sequentially, we concatenate multiple subvolumes and process them in parallel until cross-attention is reached. This approach helps reduce the running time for MLP and other linear layers in the decoder.

## D. Details of Diffusion Distillation

In Sec. 4.3, we provide a brief ablation study of the proposed progressive flow distillation with a case study. Here, we present a more detailed comparison with additional test cases and also include ablations of Huber loss and Phase 1 fine-tuning.

**Guidance Distillation Warmup.** Fig. 18 shows the results without guidance distillation warmup. We observe significant degradation in results when guidance distillation is omitted, confirming the effectiveness of our strategy.

**Huber Loss vs L2 Loss.** Fig. 19 shows a visual comparison between models trained with L2 and Huber loss. While L2 loss generates reasonable results, the quality is noticeably inferior to that of the model trained with Huber loss. For example, certain structures, like the radio and several houses, are broken in the L2 model. We hypothesize that it is because Huber loss is less sensitive to outliers, thus stabilizing the training and improving the results.

**EMA of Target Network.** Fig. 20 compares models trained with and without EMA. Both models were finetuned from a guidance-distilled model using consistency flow distillation, with no Phase 1 or adversarial fine-tuning. It can be seen that the meshes are broken without EMA, highlighting the importance of EMA for stability.

**Phase 1 Fine-tuning.** During consistency flow distillation, we follow PCM [43] to divide the total trajectory into 5 phases and force the model to predict different targets at each phase. However, there is a training-test gap as the model needs to predict final target during the inference. To address this, we propose Phase 1 fine-tuning after Phase 5 pretraining. We empirically find that this strategy slightly improves performance, as shown in Fig. 21.

**Adversarial Fine-tuning.** The comparison between models with and without adversarial fine-tuning is shown in Fig. 22. It is evident that adversarial fine-tuning helps improve surface smoothness, corrects detail generation, and fixes mesh holes.

**Effect of Sampling Steps.** As shown in Fig. 23, our method demonstrates the ability to generate rough results with just 2 steps, and simple objects can be effectively generated within 3 steps.

## E. More Results

**Shape Generation Results.** Fig. 24 presents a set of shape generation results from Hunyuan3D-2 Turbo, which has been distilled using the proposed FlashVDM framework. Our model achieves fast generation with only 5 diffusion sampling steps and ultra-fast decoding, while maintaining high-quality meshes across a variety of categories.

**Compatibility with Texture Generation.** Fig. 25 showcases texture generation results for meshes produced by Hunyuan3D-2 Turbo, distilled with FlashVDM. It is evident

that the meshes generated by our method are fully compatible with texture generation, demonstrating its versatility.

**Comparison with Other Methods.** Fig. 26 compares FlashVDM with other fast 3D generation methods. The results highlight that our method consistently outperforms existing approaches across a broad range of input types.

## F. Limitations and Future Works.

In this work, we have significantly accelerated both VAE decoding and diffusion sampling. Despite these improvements, there are still areas that could be further enhanced. For instance, our PyTorch implementation contains several indexing operations, which can slow down the GPU pipeline. Operator fusion and more efficient memory access strategies could be promising directions for optimization. Additionally, exploration of locality of vecset would also be an interesting direction. Regarding diffusion sampling, single-stage distillation may be preferable, as the current multi-stage approach is complex and introduces cascade errors, which limit its performance potential. Furthermore, while our investigation of adversarial finetuning shows promising results, further research could focus on continuously utilizing real 3D data with adversarial finetuning or even reinforcement learning, a direction we believe holds significant promise. Lastly, as VAE inference time is reduced, the proportion of time spent on diffusion sampling increases. This suggests that exploring one-step distillation could be a valuable avenue for future research.

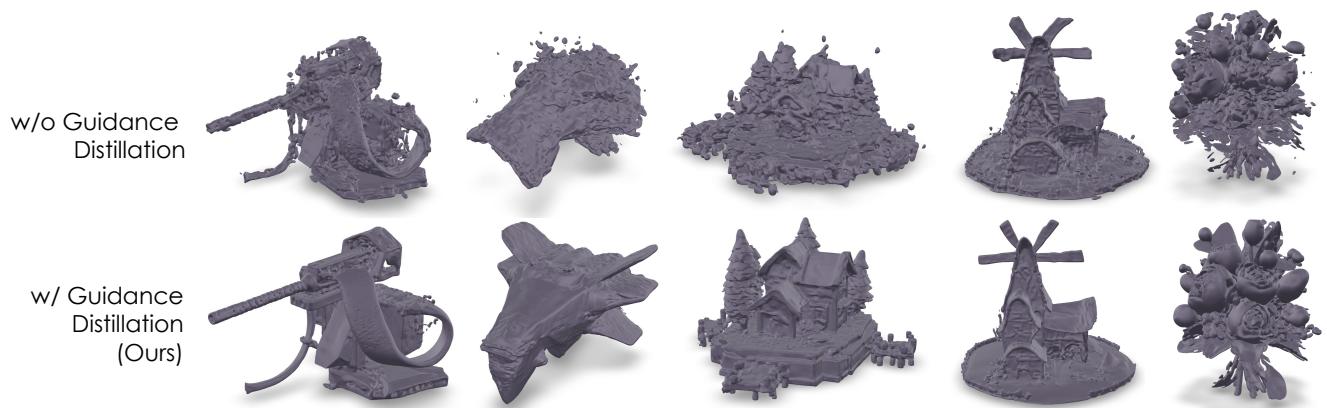


Figure 18. Visual comparison of models **with and without guidance distillation warmup**. The adversarial fine-tuning and Phase1 fine-tuning are not adopted. It demonstrates that the guidance distillation warmup is essential for successful distillation.

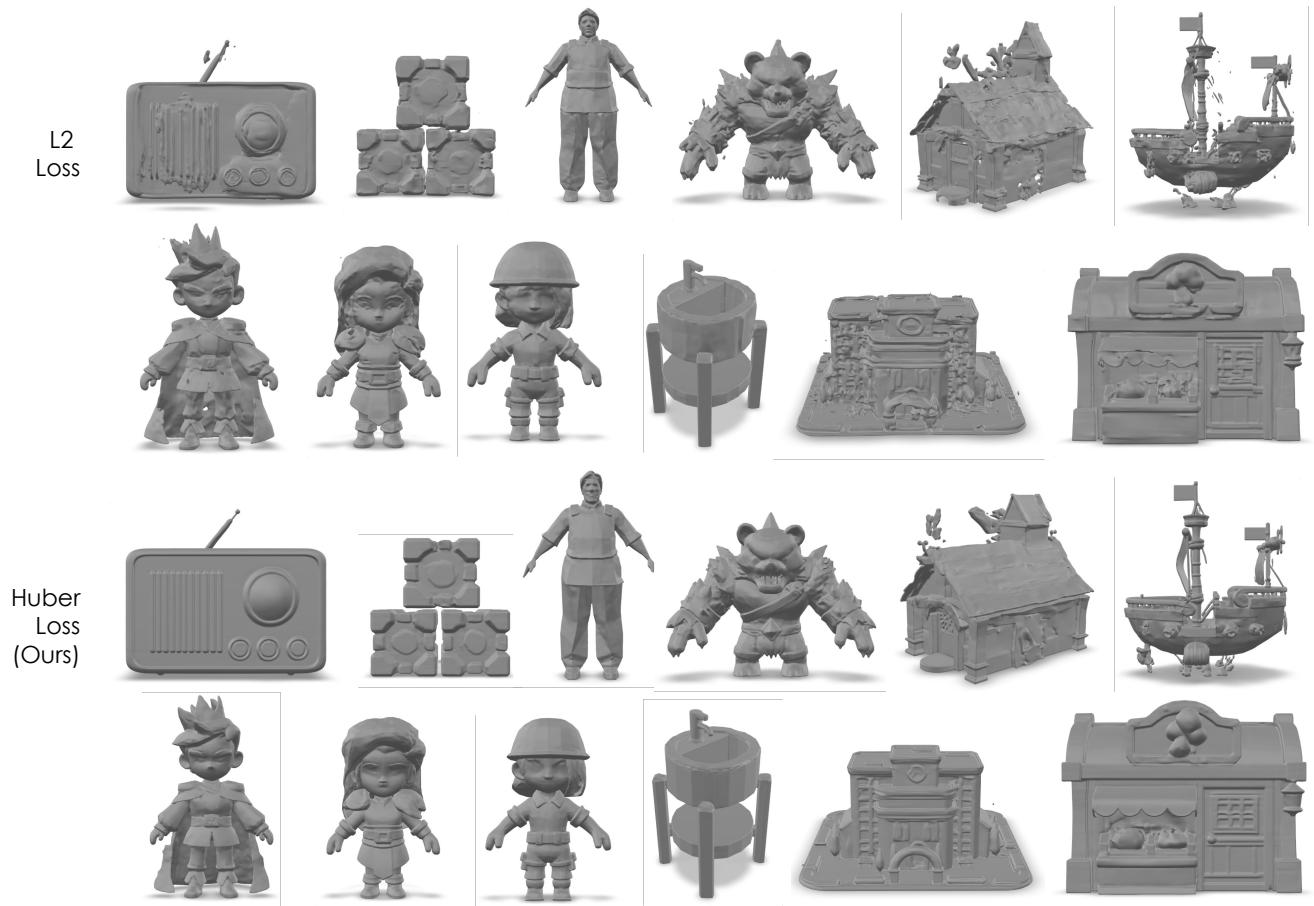


Figure 19. Visual comparison of models trained with **L2 and huber loss**. The adversarial fine-tuning and Phase1 fine-tuning are not adopted. It demonstrates that the huber loss is significantly better than l2 loss, which we hypothesis that is due to huber loss is less sensitive to outliers so that stabilizes the training and makes results better.

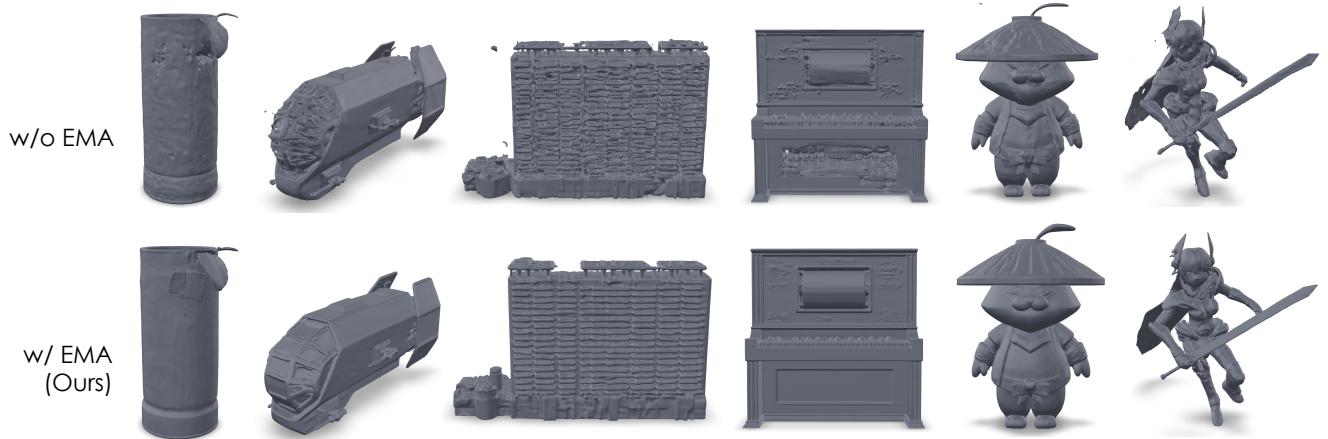


Figure 20. Visual comparison of models trained **with and without EMA**. The adversarial fine-tuning and Phase1 fine-tuning are not adopted. It demonstrates that the meshes tend to be broken without EMA.



Figure 21. Visual comparison of models **with and without guidance distillation warmup**. The adversarial fine-tuning and Phase1 fine-tuning are not adopted. It demonstrates that the guidance distillation warmup is essential for successful distillation.



Figure 22. Visual comparison of models **with and without adversarial finetuning**. All other distillation stages are used. It demonstrates that the predicted meshes are more accurate and smooth after adversarial finetuning.

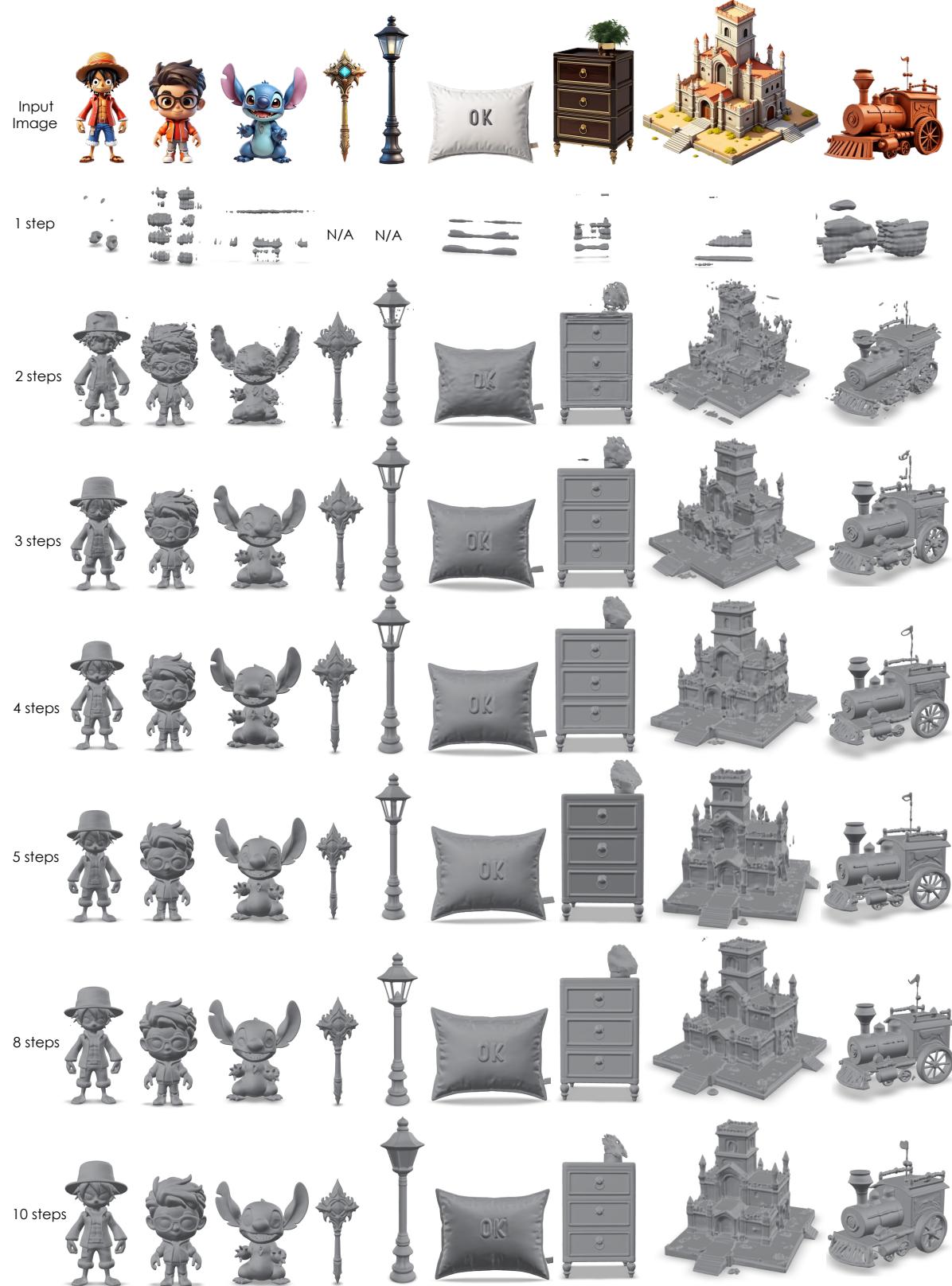


Figure 23. Visual comparison of FlashVDM generation results with different sampling steps.

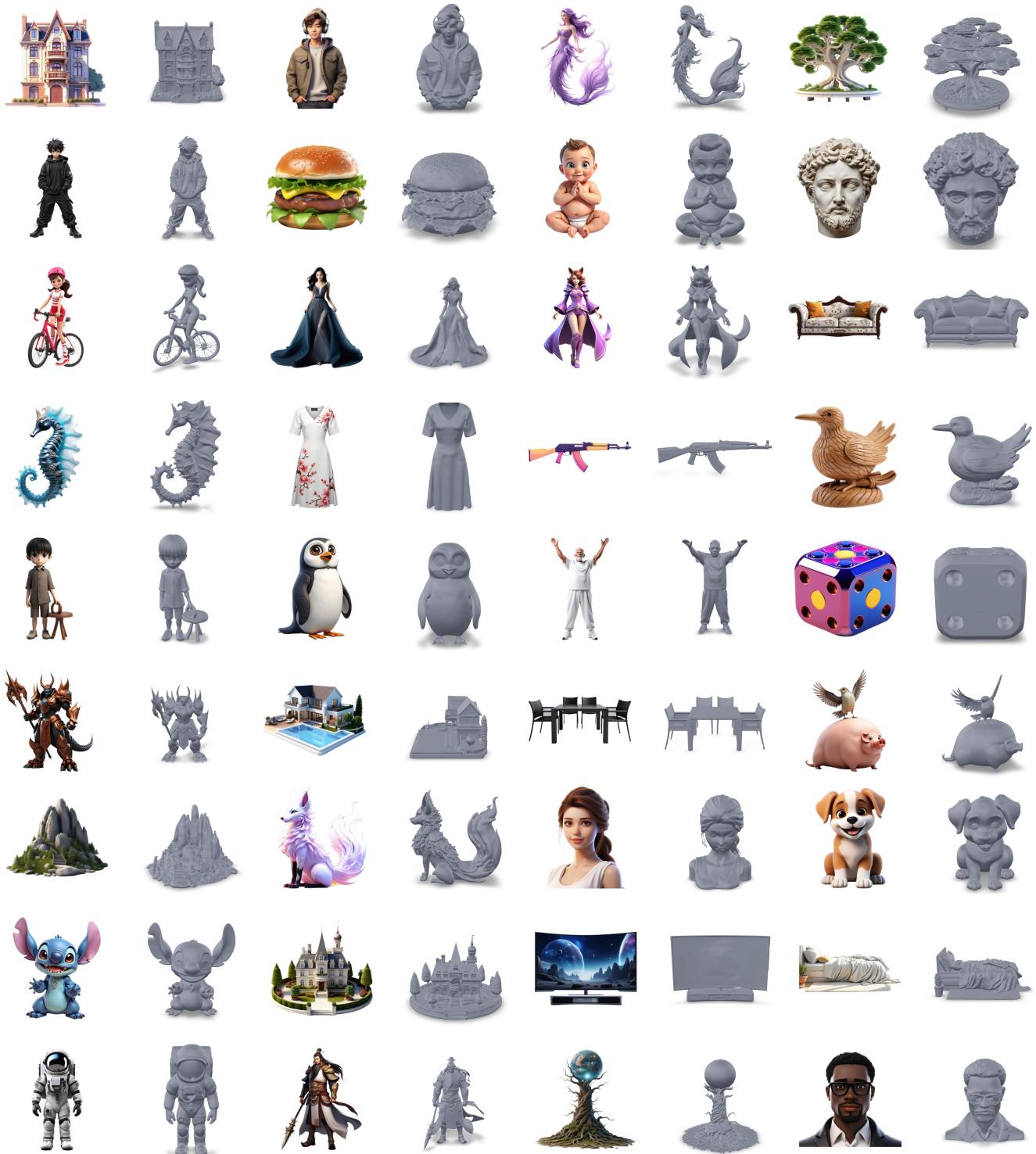


Figure 24. Shape generation results of Hunyuan3D-2 Turbo distilled with the proposed FlashVDM. Image prompts are generated by HunyuanDiT [19]. The number of inference steps is 5.

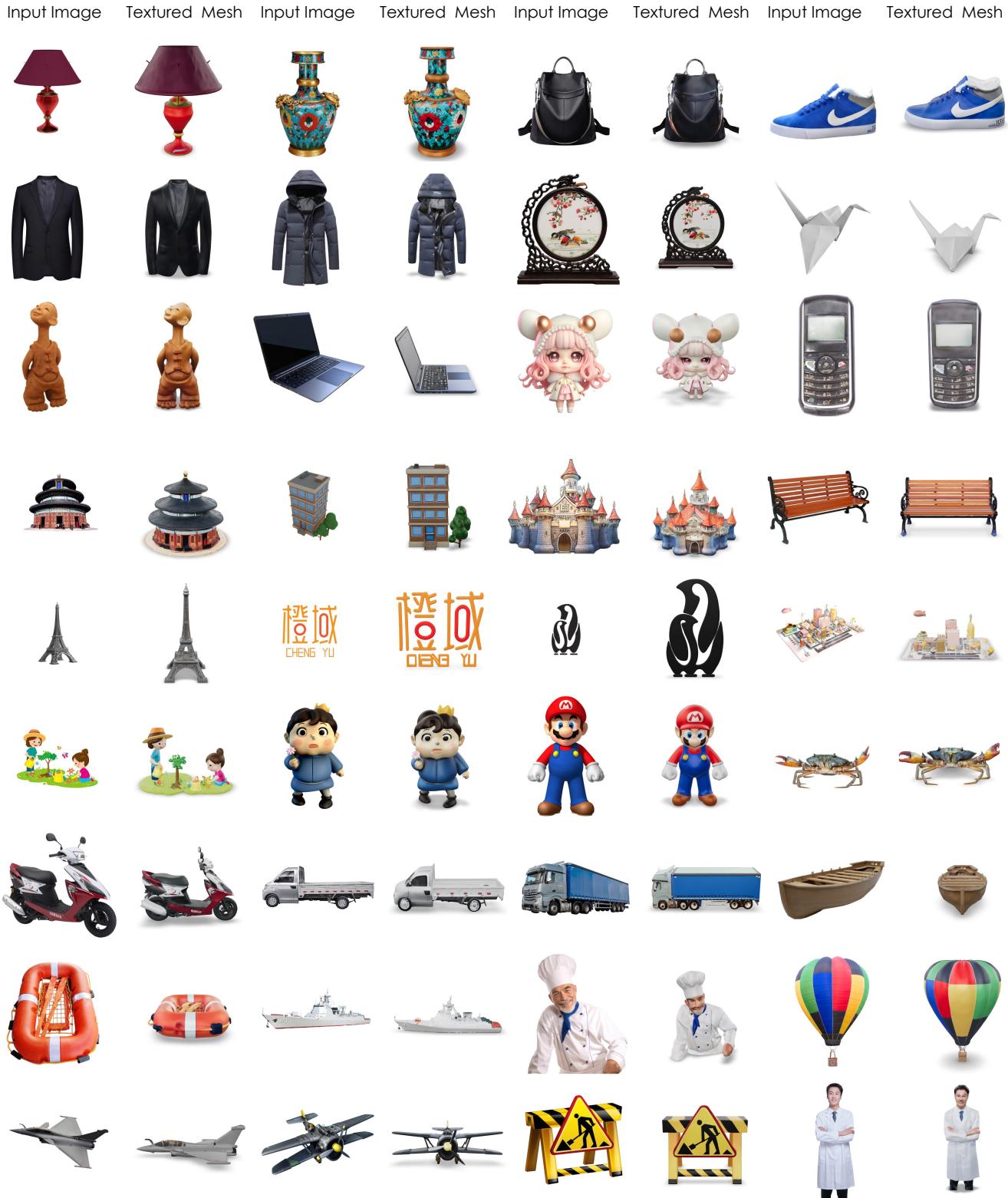


Figure 25. Texture generation results of Hunyuan3D-2 Turbo distilled with the proposed FlashVDM and Hunyuan3D-Paint-2 [57]. Image prompts are generated by HunyuanDiT [19]. The number of inference steps is 5.



Figure 26. Comparison between FlashVDM (Hunyuan3D-2 Turbo) 5 steps and other fast 3D generation methods.

## References

- [1] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023. 3
- [2] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022. 3
- [3] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint arXiv:2408.00653*, 2024. 3, 7, 8
- [4] Junyu Chen, Han Cai, Junsong Chen, Enze Xie, Shang Yang, Haotian Tang, Muyang Li, Yao Lu, and Song Han. Deep compression autoencoder for efficient high-resolution diffusion models. *arXiv preprint arXiv:2410.10733*, 2024. 3
- [5] Rui Chen, Jianfeng Zhang, Yixun Liang, Guan Luo, Weiyu Li, Jiarui Liu, Xiu Li, Xiaoxiao Long, Jiashi Feng, and Ping Tan. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. *arXiv preprint arXiv:2412.17808*, 2024. 3
- [6] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2, 4
- [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *NIPS*, 2014. 6
- [8] Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024. 3
- [9] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 3
- [10] Hanzhe Hu, Tianwei Yin, Fujun Luan, Yiwei Hu, Hao Tan, Zexiang Xu, Sai Bi, Shubham Tulsiani, and Kai Zhang. Turbo3d: Ultra-fast text-to-3d generation. *arXiv preprint arXiv:2412.04470*, 2024. 3
- [11] Zixuan Huang, Mark Boss, Aaryaman Vasishta, James M Rehg, and Varun Jampani. Spar3d: Stable point-aware reconstruction of 3d objects from single images. *arXiv preprint arXiv:2501.04689*, 2025. 3, 7, 8
- [12] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 2
- [13] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 5
- [14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with

- frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2
- [15] 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. *arXiv preprint arXiv:2311.06214*, 2023. 3
- [16] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman: High-fidelity mesh generation with 3d native generation and interactive geometry refiner, 2024. 5
- [17] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snap-fusion: Text-to-image diffusion model on mobile devices within two seconds. *Advances in Neural Information Processing Systems*, 36:20662–20678, 2023. 3
- [18] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Tripogs: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025. 2, 9
- [19] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiaxin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. 5, 17, 18
- [20] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 6
- [21] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9298–9309, 2023. 3
- [22] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 3, 4
- [23] Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*, 2024. 2
- [24] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 3
- [25] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 3
- [26] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 2, 5, 10
- [27] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024. 3
- [28] Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *CVPR*, 2023. 2, 3, 10
- [29] Maxime Oquab, Timothée Darcret, 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, 2023. 3
- [30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 3
- [31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 5
- [32] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Moriguchi, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2304–2314, 2019. 3, 4
- [33] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022. 3
- [34] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023. 2, 3
- [35] Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. *arXiv preprint arXiv:2403.12015*, 2024. 2, 3, 6
- [36] 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. 3
- [37] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3
- [38] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023. 2, 6

- [39] Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. In *ICLR*, 2024. 3
- [40] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023. 2, 3, 5
- [41] 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. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 3
- [42] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforet, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 3, 7, 8
- [43] Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in Neural Information Processing Systems*, 37:83951–84009, 2025. 2, 3, 5, 6, 10, 12
- [44] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *ICLR*, 2023. 3
- [45] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*, 2024. 6, 7
- [46] 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. *arXiv preprint arXiv:2412.01506*, 2024. 2, 8
- [47] Chen Xu, Tianhui Song, Weixin Feng, Xubin Li, Tiezheng Ge, Bo Zheng, and Limin Wang. Accelerating image generation with sub-path linear approximation model. *arXiv preprint arXiv:2404.13903*, 2024. 3
- [48] Yanwu Xu, Yang Zhao, Zhisheng Xiao, and Tingbo Hou. Ufogen: You forward once large scale text-to-image generation via diffusion gans. In *CVPR*, 2024. 3, 6
- [49] Le Xue, Mingfei Gao, Chen Xing, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and Silvio Savarese. Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1179–1189, 2023. 7
- [50] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. *arXiv preprint arXiv:2405.07510*, 2024. 3
- [51] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédéric Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. *arXiv preprint arXiv:2405.14867*, 2024. 6
- [52] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédéric Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024. 2, 3
- [53] Yuanhao Zhai, Kevin Lin, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Chung-Ching Lin, David Doermann, Junsong Yuan, and Lijuan Wang. Motion consistency model: Accelerating video diffusion with disentangled motion-appearance distillation. *Advances in Neural Information Processing Systems*, 37:111000–111021, 2025. 2
- [54] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023. 2, 3, 5, 6, 7
- [55] Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization, 2024. 3
- [56] 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. 2, 3
- [57] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *ICLR*, 2021. 18
- [58] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5, 6, 7
- [59] 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*, 2025. 1, 2, 3, 6, 7, 8, 9, 10
- [60] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*, 2023. 7
- [61] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024. 3
- [62] Yuanzhi Zhu, Hanshu Yan, Huan Yang, Kai Zhang, and Junnan Li. Accelerating video diffusion models via distribution matching. *arXiv preprint arXiv:2412.05899*, 2024. 2, 5