Few-step Flow for 3D Generation via Marginal-Data Transport Distillation

Zanwei Zhou[*1], Taoran Yi[*2], Jiemin Fang[†3], Chen Yang[3], Lingxi Xie[3], Xinggang Wang[2], Wei Shen[†1], Qi Tian[3]

[1]MoE Key Lab of Artificial Intelligence, AI Institute, School of Computer Science, Shanghai Jiao Tong University
[2]School of EIC, Huazhong University of Science and Technology
[3]Huawei Inc.

Abstract

Flow-based 3D generation models typically require dozens of sampling steps during inference. Though few-step distillation methods, particularly Consistency Models (CMs), have achieved substantial advancements in accelerating 2D diffusion models, they remain under-explored for more complex 3D generation tasks. In this study, we propose a novel framework, **MDT-dist**, for few-step 3D flow distillation. Our approach is built upon a primary objective: **dist**illing the pretrained model to learn the **M**arginal-**D**ata **T**ransport. Directly learning this objective needs to integrate the velocity fields, while this integral is intractable to be implemented. Therefore, we propose two optimizable objectives, Velocity Matching (VM) and Velocity Distillation (VD), to equivalently convert the optimization target from the transport level to the velocity and the distribution level respectively. Velocity Matching (VM) learns to stably match the velocity fields between the student and the teacher, but inevitably provides biased gradient estimates. Velocity Distillation (VD) further enhances the optimization process by leveraging the learned velocity fields to perform probability density distillation. When evaluated on the pioneer 3D generation framework TRELLIS, our method reduces sampling steps of each flow transformer from 25 to 1–2, achieving **0.68**s (1 step $\times$ 2) and **0.94**s (2 steps $\times$ 2) latency with **9.0**$\times$ and **6.5**$\times$ speedup on A800, while preserving high visual and geometric fidelity. Experiments demonstrate that our method significantly outperforms existing CM distillation methods, and enables TRELLIS to achieve superior performance in few-step 3D generation.

Code — https://github.com/Zanue/MDT-dist

Project Page — https://zanue.github.io/mdt-dist

Introduction

Flow-based 3D generation models (Xiang et al. 2024; Zhao et al. 2025; Yang et al. 2024; Li et al. 2025; Wu et al. 2024, 2025; Chen et al. 2025b; Ye et al. 2025; Zhang et al. 2024b) have exhibited remarkable abilities in synthesizing intricate 3D representations from image prompts. However, during inference they typically require dozens of iterative sampling steps, posing significant computational barriers to practical applications such as large-scale 3D content generation for embodied intelligence simulation (Wang et al. 2025) and real-time interactive editing workflows in graphics systems. Although few-step diffusion distillation methods, particularly consistency models (CMs) (Song et al. 2023; Wang et al. 2024; Lu and Song 2024), have achieved substantial advancements in accelerating 2D diffusion models, their extension to the 3D generation area remains underexplored. The only related work recently is FlashVDM (Lai et al. 2025), which adopts a few-step distillation framework mainly derived from the previous Phased Consistency Models (PCM) method (Wang et al. 2024).

3D generation presents inherently greater challenges than its 2D counterpart. Unlike 2D images sampled from a continuous color space, 3D representations, e.g., meshes and 3D Gaussians (Kerbl et al. 2023), are discrete and sparsely structured in 3D space. 3D models also contain richer geometric and textural details at a finer granularity. Moreover, in latent-space generative frameworks such as Latent Diffusion models (LDM) (Rombach et al. 2022), the dimension of the 3D latent space is typically higher than that of the 2D latent space. These fundamental differences indicate that 3D generation faces more difficulties and challenges than 2D generation, and thus has stricter requirements on few-step acceleration techniques.

To address these challenges, we propose a novel framework, MDT-dist, for few-step 3D flow distillation. Our method is built upon a primary objective: distilling the pretrained 3D flow model to learn the marginal-data transport. CMs have a similar optimization target, but are limited by the consistency constraint which enforces consistency between adjacent time steps to indirectly learn the target. We instead propose two novel loss functions, Velocity Matching (VM) and Velocity Distillation (VD), to pursue the primary objective in a more direct way. Directly learning the primary objective needs to integrate the velocity fields, but this integral is intractable to be implemented. Therefore, VM and VD equivalently convert it to tractable objectives respectively. In VM, the optimization of the primary objective is converted into optimizing its time derivative, with the error in the primary objective bounded by the error in the VM loss. Specifically, VM learns to stably match the velocity fields between the student and the teacher. However, it inevitably contains a term involving the derivative of the network out-

[*]Equal contribution. Work done during internship at Huawei.
[†]Corresponding Author.

put. This term cannot be efficiently back-propagated and is therefore detached, leading to biased gradient estimates. VD further enhances the learning of the marginal-data transport by matching the marginal distributions between the student and the teacher. It leverages the velocity fields learned by the student and the teacher as measure to perform probability density distillation. We evaluate our methods on a state-of-the-art 3D generation framework TRELLIS (Xiang et al. 2024). Our approach reduces the inference steps of each flow transformer from 25 to just 1-2 and the latency from 6.1s to 0.68s and 0.94s on A800, while preserving high visual and geometric fidelity. Extensive experiments demonstrate that our method significantly outperforms existing CM distillation methods, and makes the distilled TRELLIS model surpass FlashVDM (Lai et al. 2025), enabling fast 3D content generation beneficial for various downstream tasks.

Our contributions are as follows:

- We develop a novel few-step flow distillation framework MDT-dist for better 3D generation acceleration.
- We propose two novel optimization objectives, Velocity Matching (VM) and Velocity Distillation (VD), to jointly enable effective few-step distillation.
- We distilled TRELLIS to achieve a sweet balance between generation speed and quality. Our method reduces sampling steps of each flow transformer from 25 to 1–2, achieving 0.68s (1 step $\times$ 2) and 0.94s (2 steps $\times$ 2) latency with 9.0$\times$ and 6.5$\times$ speedup on A800.

Related Work

3D Generation Models. Early 3D generation methods (Jain et al. 2022; Lin et al. 2023; Shi et al. 2023; Raj et al. 2023; Gao et al. 2025; Liu et al. 2023; Long et al. 2023) are mainly based on 2D diffusion models (Rombach et al. 2022), generating 3D assets by iteratively prompting 2D diffusion models to optimize 3D representations (Mildenhall et al. 2020; Kerbl et al. 2023). DreamFusion (Poole et al. 2022) and Score Jacobian Chaining (SJC) (Song et al. 2020) first introduce Score Distillation Sampling (SDS) to generate 3D assets using pretrained 2D diffusion models. ProlificDreamer (Wang et al. 2023) and other methods (Liang et al. 2023; Sun et al. 2023; Zhao et al. 2023) further improve SDS to achieve better generation results. Some methods (Yi et al. 2023; Tang et al. 2023; Yi et al. 2024) incorporate shape priors to significantly reduce generation time. Methods (Gupta et al. 2023; Shen et al. 2024; Li et al. 2023; Hong et al. 2023; Tang et al. 2024; Xu et al. 2024; Zhang et al. 2024a) like LRM (Hong et al. 2023) and LGM (Tang et al. 2024) build native 3D generative models by pretraining on large-scale 3D data, enabling feed-forward generation of 3D assets without optimization. Some native 3D generation methods (Xiang et al. 2024; Zhao et al. 2025; Yang et al. 2024; Li et al. 2025; Wu et al. 2024, 2025; Chen et al. 2025c; Ye et al. 2025; Chen et al. 2025a; Zhang et al. 2024b; Li et al. 2024) further introduce flow matching into the 3D generation field. However, generating high-fidelity 3D assets with 3D diffusion models requires a relatively large number of sampling steps during inference, which both increases users' queuing time and raises computational costs. FlashVDM (Lai et al. 2025) shortens the generation time through efficient decoder design and few-step distillation, but it can only generate the shape without appearance. We apply our method to TRELLIS (Xiang et al. 2024) to reduce the number of sampling steps for the two stage flow transformers, enabling the fast generation on both shape and appearance.

2D Generative Models. 2D generative modeling has progressed from variational autoencoders (VAEs) (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014) to generative adversarial networks (GANs) (Goodfellow et al. 2020). VAEs map data distributions to latent Gaussian spaces via Evidence Lower Bound (ELBO) optimization but often yield blurry outputs. GANs employ adversarial training to produce high-fidelity images but suffer from instability and mode collapse. Diffusion models have since become a dominant paradigm: denoising diffusion probabilistic models (DDPM) (Ho, Jain, and Abbeel 2020) formalize iterative denoising as a Markov process; score-based generative modeling (Song et al. 2020) unifies diffusion under the framework of stochastic differential equations (SDEs); flow matching (Lipman et al. 2022; Liu, Gong, and Liu 2022; Tong et al. 2023) learns velocity fields for direct ODE-based generation and often requires fewer steps than diffusion. An important topic on diffusion models is acceleration by reducing the number of sample steps. Denoising diffusion implicit models (DDIM) (Song, Meng, and Ermon 2020) accelerate generation through non-Markovian updates. DPM-Solver (Lu et al. 2022) further reduces sampling steps by employing higher-order ODE solvers. Consistency models (CMs) (Song et al. 2023) enable single-step sampling by enforcing consistency across sampling trajectories. Variants such as phased consistency models (PCM) (Wang et al. 2024) and Trigflow(sCM) (Lu and Song 2024) improve stability through phased training strategies and continuous-time formulations, respectively. In contrast to consistency models, our method is derived without the consistency constraint and serves as a novel few-step distillation framework.

Background

Diffusion Models

Diffusion models (Ho, Jain, and Abbeel 2020) learn to generate data by iteratively denoising samples from Gaussian noise distribution. The framework defines a fixed forward diffusion process and a learned reverse denoising process. Given a data sample $\boldsymbol{x}_0 \sim q_{\text{data}}$, the forward diffusion process gradually adds the Gaussian noise and produces a series of noised samples $\{\boldsymbol{x}_t\}_{t=1}^{T}$, conditioned on the time step t. This induces a sequence of marginal distributions $q_t(\boldsymbol{x}_t)$, i.e., the noised distribution at time t. The reverse process is parameterized by a neural network with learnable parameters θ, which learns to generate the denoising direction. During the reverse process, the generated marginal distribution $p_\theta^t(\boldsymbol{x}_t)$ is expected to be matched with $q_t(\boldsymbol{x}_t)$.

Flow Matching. Flow matching (FM) (Lipman et al. 2022; Liu, Gong, and Liu 2022; Tong et al. 2023) learns to map noise to data distribution by estimating a Probability Flow Ordinary Differential Equation (PF-ODE) process.

It defines a continuous-time dynamical system with a learnable velocity field $\boldsymbol{v}_\theta(\cdot, t), t \in [0, 1]$, which can be used to construct a time-dependent diffeomorphic map $\boldsymbol{\phi}_t$,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi}_t(\boldsymbol{x}) &= \boldsymbol{v}_\theta(\boldsymbol{\phi}_t(\boldsymbol{x}), t), \\ \boldsymbol{\phi}_0(\boldsymbol{x}) &= \boldsymbol{x}, \end{aligned} \tag{1}$$

which subsequently defines a push-forward $\boldsymbol{\phi}_*$ transforming a density over time

$$p_\theta^t(\boldsymbol{x}) = [\boldsymbol{\phi}_t]_* p_0(\boldsymbol{x}) = p_0(\boldsymbol{\phi}^{-1}(\boldsymbol{x})) \left|\det \nabla_{\boldsymbol{x}} \boldsymbol{\phi}_t^{-1}(\boldsymbol{x})\right|. \tag{2}$$

In this way, the velocity field $\boldsymbol{v}_\theta(\cdot, t)$ is said to generate a probability path p_θ^t. The velocity field is optimized by minimizing the conditional flow matching loss

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{z}} \left[\left\| \boldsymbol{v}_\theta\left((1-t)\boldsymbol{x}_0 + t\boldsymbol{z}, t\right) - (\boldsymbol{z} - \boldsymbol{x}_0) \right\|^2 \right]. \tag{3}$$

3D Generation Models

Our method builds upon TRELLIS (Xiang et al. 2024), a recent high-quality 3D asset generation framework. In TRELLIS, a 3D asset is implicitly represented by a structured latent variable (SLAT) $\boldsymbol{S}$, which is composed of sparse voxels and features:

$$\boldsymbol{S} = \{(f_i, \boldsymbol{p}_i)\}_{i=1}^L, \quad f_i \in \mathbb{R}^C,\ \boldsymbol{p}_i \in \{0, 1, \ldots, N-1\}^3, \tag{4}$$

where $\boldsymbol{p}_i$ is the coordinate of the i-th voxel, and f_i is the corresponding feature. C denotes the feature dimension, N denotes the voxel resolution, and L denotes the number of active voxels which is much smaller than N^3. SLAT can be decoded into different 3D representations such as 3D Gaussians (Kerbl et al. 2023), meshes and NeRFs (Mildenhall et al. 2020). For generation, two flow transformers are trained to generate coordinates and features of SLAT separately. During inference, the two models both take 25 sampling steps by default.

Marginal-Data Transport Distillation

Primary Objective

Given a noise sample $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, our target is to learn a neural network $\boldsymbol{\phi}_\theta$ to generate $\boldsymbol{x}_0 \sim q_{\text{data}}$ within few-step forwards. Taking one-step generation as an example, $\boldsymbol{\phi}_\theta$ is expected to satisfy

$$\min_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{z}} \left[D\left(\boldsymbol{\phi}_\theta(\boldsymbol{z}), \boldsymbol{z} - \boldsymbol{x}_0\right) \right], \tag{5}$$

where $D(\cdot, \cdot)$ represents a distance metric such as MSE or Kullback-Leibler Divergence, and θ represents learnable model parameters. This formulation aims to transport the noise distribution to the data distribution directly. However, from the experiences of VAEs (Kingma and Welling 2013) and GANs (Goodfellow et al. 2020), it is well known that directly learning data distribution from noise distribution is a challenging task and easily leads to unstable optimization issues such as posterior or mode collapse. Therefore, we instead force $\boldsymbol{\phi}_\theta$ to learn the transport from the marginal distribution $q_t(\boldsymbol{x}_t)$ to the data distribution q_{data} for any diffusion time step t:

$$\min_\theta \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{z}} \left[D\left(t\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t), \boldsymbol{x}_t - \boldsymbol{x}_0\right) \right], \tag{6}$$

where t is used for normalization, $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{z}$. With the guidance of a well-pretrained 3D flow model, we can approximate $q_t(\boldsymbol{x}_t)$ using its learned reverse marginal distribution, and approximate the data distribution q_{data} using the generated teacher distribution. The transport from $p_\theta^t(\boldsymbol{x}_t)$ to p_0 is $\int_0^t \boldsymbol{v}_\theta(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau$, thus our *primary objective* is formulated as

$$\begin{aligned} &\mathcal{L}_{\text{primary}}(\theta) \coloneqq \\ &\min_\theta \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{z}} \left[D\left(t\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t), \int_0^t \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau\right)\right], \end{aligned} \tag{7}$$

where $\boldsymbol{v}_{\text{pretrain}}$ denotes the velocity fields predicted by the pretrained teacher model.

Velocity Matching

Note that the primary objective in Eq. 7 cannot be directly optimized, since the integral $\int_0^t \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_\tau, \tau)\mathrm{d}\tau$ is intractable. We differentiate the objective function with respect to t and turn it to be

$$\min_\theta \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{z}} \left[D\left(\boldsymbol{u}_\theta(\boldsymbol{x}_t, t), \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_t, t)\right)\right], \tag{8}$$

$$\boldsymbol{u}_\theta(\boldsymbol{x}_t, t) = \boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t) + t\frac{\mathrm{d}\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t)}{\mathrm{d}t}. \tag{9}$$

Here $\boldsymbol{u}_\theta(\boldsymbol{x}_t, t)$ is the derivative of $t\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t)$ with respect to t, which actually represents the velocity fields. Intuitively, Eq. 7 supervises $\boldsymbol{\phi}_\theta$ by the transport from the marginal distribution to the data distribution, and Eq. 8 converts it to be a supervision on its derivative, i.e., the velocity supervision, to learn student velocity fields matched with the teacher velocity fields.

Now Eq. 8 is tractable and can be directly used for supervision. Given a data sample $\boldsymbol{x}_0 \sim q_{\text{data}}$, we replace $D(\cdot, \cdot)$ with the MSE metric and define our velocity matching loss as

$$\begin{aligned} \mathcal{L}_{\text{VM}}(\theta) \coloneqq \min_\theta \mathbb{E}_{t, \boldsymbol{x}_0, \boldsymbol{z}} \Big[\Big\| \boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t) + t\frac{\mathrm{d}\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t)}{\mathrm{d}t} - \\ \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_t, t) \Big\|^2 \Big]. \end{aligned} \tag{10}$$

We provide a detailed proof in Appendix to demonstrate that, the error in the primary objective $\mathcal{L}_{\text{primary}}(\theta)$ is bounded by the error in the velocity matching loss $\mathcal{L}_{\text{VM}}(\theta)$. Therefore, we can effectively learn the primary objective through optimizing the velocity matching loss.

In practice, we accelerate the convergence by discretely approximating the derivative item

$$\frac{\mathrm{d}\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t)}{\mathrm{d}t} \approx \frac{1}{\Delta t}\left(\boldsymbol{\phi}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{\phi}_\theta(\boldsymbol{x}_{t-\Delta t}, t - \Delta t)\right), \tag{11}$$

where $\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_t - \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_t, t)\Delta t$, Δt is a small constant value and we set it to be $1e-2$. The gradient of the

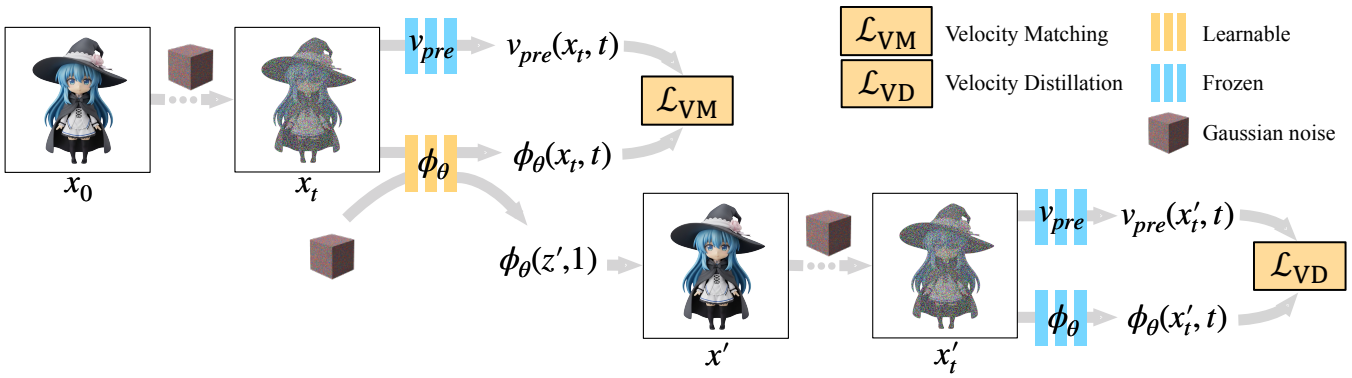

Figure 1: The primary objective of our framework is to learn the transport from the marginal distribution to the data distribution. Based on it we propose two optimization objectives: velocity matching and velocity distillation. Velocity matching directly supervises ϕ_θ via matching the velocity fields between the student and teacher (Eq. 10), while velocity distillation indirectly supervises ϕ_θ via matching the marginal distributions between the student and teacher (Eq. 14).

term $\phi_\theta(\boldsymbol{x}_{t-\Delta t}, t-\Delta t)$ is expected to be detached, since solving it requires computing two-order derivative which is computationally expensive. When training, we stop the gradient of the derivative term $\frac{\mathrm{d}\phi_\theta(\boldsymbol{x}_t,t)}{\mathrm{d}t}$ rather than only $\phi_\theta(\boldsymbol{x}_{t-\Delta t}, t-\Delta t)$. This operation makes $\phi_\theta(\boldsymbol{x}_t, t)$ to learn more consistent with $\boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_t, t)$, achieving a more stable optimization process.

Velocity Distillation

Though Eq. 10 serves as a tractable objective, we clarify that with only the velocity matching loss the student model ϕ_θ cannot be optimized well. An inherent flaw in the velocity matching loss is that both the continuous and discrete formulation of $\phi_\theta(\boldsymbol{x}_t, t)$ cannot be back-propagated properly. Therefore, the optimization of Eq. 10 provides a biased gradient estimate.

We revisit our primary objective from the perspective of score distillation (Poole et al. 2022; Wang et al. 2023). Since the student and teacher model have the same target distribution q_{data}, we can equivalently convert the constraint on the marginal-data transport into the constraint on the marginal distribution. Then the primary objective turns to minimize the Kullback-Leibler divergence between the student marginal p_θ^t and the marginal q_t (note we approximate the real marginal with the teacher marginal):

$$\min_\theta \mathbb{E}_t \left[D_{\text{KL}} \left(p_\theta^t \parallel q_t \right) \right], \tag{12}$$

which is the objective of our velocity distillation. Specifically, a sample $\boldsymbol{x}' = \boldsymbol{z}' - \phi_\theta(\boldsymbol{z}')$ is first synthesized from a noise sample $\boldsymbol{z}' \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. Then $\boldsymbol{x}'$ is diffused with $\boldsymbol{z}'' \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ to obtain $\boldsymbol{x}'_t = (1-t)\boldsymbol{x}' + t\boldsymbol{z}''$. The gradient of the training objective can then be written as

$$\begin{aligned}
&\nabla_\theta \mathbb{E}_t \left[D_{\text{KL}} \left(p_\theta^t \parallel q_t \right) \right] \\
&= \mathbb{E}_{t,\boldsymbol{z}',\boldsymbol{z}''} \left[\nabla_\theta \left(\log p_\theta^t(\boldsymbol{x}'_t) - \log q_t(\boldsymbol{x}'_t) \right) \right] \\
&= \mathbb{E}_{t,\boldsymbol{z}',\boldsymbol{z}''} \left[\left(\nabla_{\boldsymbol{x}'_t} \log p_\theta^t(\boldsymbol{x}'_t) - \nabla_{\boldsymbol{x}'_t} \log q_t(\boldsymbol{x}'_t) \right) \frac{\partial \boldsymbol{x}'_t}{\partial \theta} \right].
\end{aligned} \tag{13}$$

ProlificDreamer (Wang et al. 2023) points out that $\nabla_{\boldsymbol{x}'_t} \log p_\theta^t(\boldsymbol{x}'_t)$ and $\nabla_{\boldsymbol{x}'_t} \log q_t(\boldsymbol{x}'_t)$ represent the score (Song et al. 2020) of the noisy prediction and the noisy real data respectively. Here we use the student velocity fields $\boldsymbol{u}_\theta(\boldsymbol{x}'_t, t)$ to replace $-\nabla_{\boldsymbol{x}'_t} \log p_\theta^t(\boldsymbol{x}'_t)$, and the teacher velocity fields $\boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}'_t, t)$ to replace $-\nabla_{\boldsymbol{x}'_t} \log q_t(\boldsymbol{x}'_t)$. A detailed proof of the rationale for this choice can be found in Appendix. Substituted with Eq. 9, the gradient of our velocity distillation loss is formulated as

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{\text{VD}}(\theta) \coloneqq \mathbb{E}_{t,\boldsymbol{z}',\boldsymbol{z}''} \Big[-\Big(\boldsymbol{\phi}_\theta(\boldsymbol{x}'_t, t) + t\frac{\mathrm{d}\boldsymbol{\phi}_\theta(\boldsymbol{x}'_t, t)}{\mathrm{d}t} - \\
\boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}'_t, t) \Big) \frac{\partial \boldsymbol{x}'_t}{\partial \theta} \Big].
\end{aligned} \tag{14}$$

Intuitively, Eq. 14 performs probability density distillation which serves as a soft supervision. It applies $\boldsymbol{u}_\theta(\boldsymbol{x}'_t, t) - \boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}'_t, t)$ as a criterion to measure the discrepancy between the teacher and student marginals, and indirectly optimizes ϕ_θ through optimizing $\boldsymbol{x}'_t$.

Optimization

Our velocity matching loss and velocity distillation loss are complementary. While both are d erived from the primary objective, velocity matching provides partially right gradient estimates, improving the performance of $\phi_\theta(\cdot, t)$ for all the time steps t. Velocity distillation further utilizes the optimized $\phi_\theta(\cdot, t)$ as a criterion for measuring distribution discrepancy, enhancing the one-step performance of ϕ_θ. Our final loss function is formulated as

$$\mathcal{L}_{\text{MDT-dist}} = \mathcal{L}_{\text{VM}} + \lambda \mathcal{L}_{\text{VD}}, \tag{15}$$

where λ is a hyper-parameter and we set it to be 1.0.

Relation to Prior Work

Velocity matching and velocity distillation are both derived from our primary objective. Velocity matching is related to consistency models (Song et al. 2023) and MeanFlow (Geng et al. 2025), but still has essential differences.

Method	Inference Steps	Inference Time (s)	Appearance $\text{FD}_{\text{incep}}\downarrow$	Appearance $\text{FD}_{\text{dinov2}}\downarrow$	Geometry $\text{ULIP}_{\text{I}}\uparrow$
LGM*	–	5	26.31	322.71	–
3DTopia-XL*	25	5	37.68	437.37	–
Ln3Diff*	250	8	26.61	357.93	–
TRELLIS*	25×2	–	9.35	67.21	–
TRELLIS	25×2	6.1	11.80	65.24	39.53
FlashVDM	5	1.30	–	–	37.91
Ours	1×2	0.68	18.09	164.2	36.88
	2×2	0.94	14.16	110.9	39.11

Table 1: Quantitative comparison on LGM (Tang et al. 2024), 3DTopia-XL (Chen et al. 2025c), Ln3Diff (Lan et al. 2024), and our teacher model TRELLIS (Xiang et al. 2024). * denotes that the metrics are from TRELLIS, which are measured on the subset of the Toys4K dataset (Stojanov, Thai, and Rehg 2021), and the inference time comes from their original paper. The other reported inference times measured by us are based on evaluations performed on an NVIDIA A800 GPU.

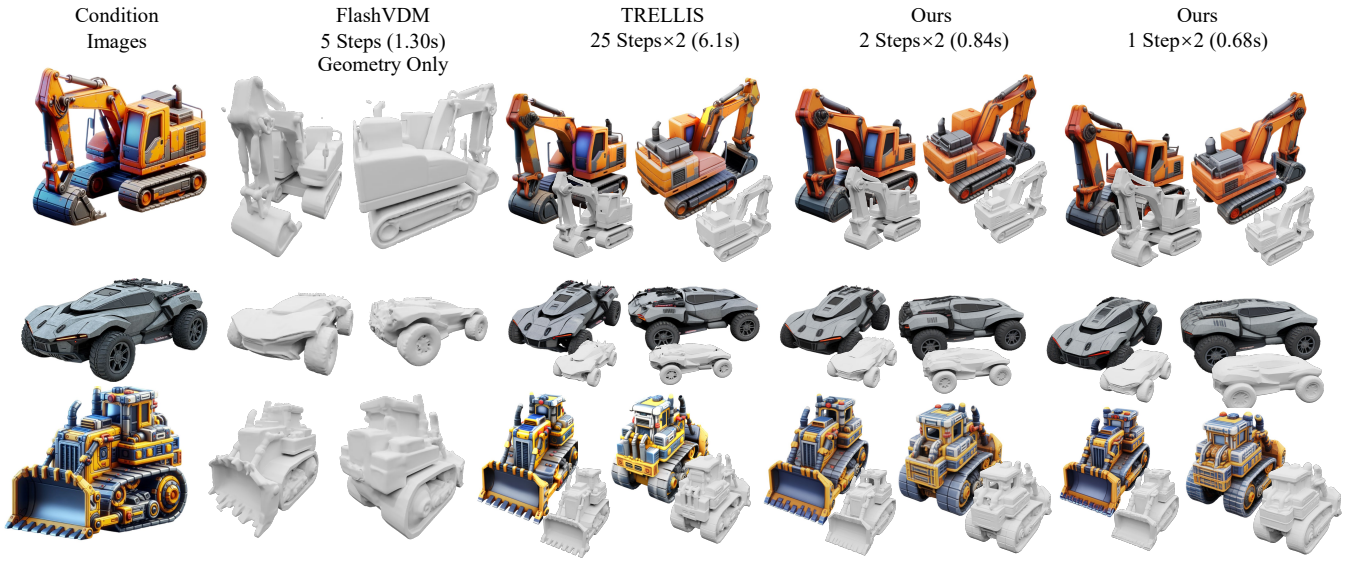

Figure 2: Qualitative results of FlashVDM (Lai et al. 2025), the teacher model TRELLIS (Xiang et al. 2024), and our method. Since FlashVDM does not generate the appearance of 3D assets, we compare with FlashVDM only on geometry.

Consistency models are derived from the consistency constraint between adjacent time steps. Compared with consistency models, our objective learns $\phi_\theta(\boldsymbol{x}_t, t)$ more consistent with $\boldsymbol{v}_{\text{pretrain}}(\boldsymbol{x}_t, t)$, leading to more stable optimization. MeanFlow needs two time variables as model input, thus being not suitable for distilling a pretrained 3D flow model. Differently, our method is motivated by learning the marginal-data transport, and is designed to distill a pretrained flow model.

Velocity distillation is related to Score Distillation Sampling (SDS) (Poole et al. 2022) and Variational Score Distillation (VSD) (Wang et al. 2023). SDS and VSD both try to measure the student and teacher marginals with the score, i.e., the gradient of the log probability density. SDS only uses the added noise as the student score, leading to a single-point Dirac distribution estimation (Wang et al. 2023). VSD finetunes an additional diffusion model ϵ_ϕ to learn the student distribution, resulting in additional memory cost and further learning errors. Instead, our method only learns one model to be both the generator and the distribution measure, being both low-cost and accurate.

Experiments

Datasets. For training, we use around 500K 3D assets from the Objaverse (XL) (Deitke et al. 2023), ABO (Collins et al. 2022), 3D-FUTURE (Fu et al. 2021), and HSSD datasets (Khanna* et al. 2023). For evaluation, we use all the 3,218 3D assets from the Toys4k dataset (Stojanov, Thai, and Rehg 2021).

Figure 3: Qualitative results with and without distillation during few-step inference.

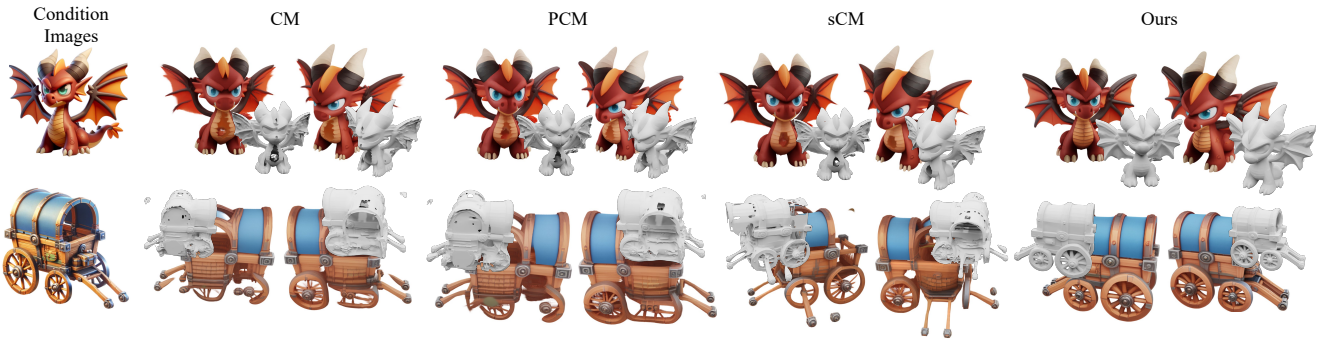

Figure 4: Qualitative comparison on CM (Song et al. 2023), PCM (Wang et al. 2024), sCM (Lu and Song 2024) and ours. Our method exhibits the most complete and fine-grained geometric and visual details.

	$\mathbf{FD_{incep}}\downarrow$	$\mathbf{FD_{dinov2}}\downarrow$	$\mathbf{ULIP_I}\uparrow$
CM	20.06	194.5	34.62
PCM	19.53	189.5	34.96
sCM	19.29	186.0	35.04
Ours	18.09	164.2	36.88

Table 2: Quantitative comparison on CM (Song et al. 2023), PCM (Wang et al. 2024), sCM (Lu and Song 2024) and ours.

$\mathcal{L}_{\text{VM}}$	$\mathcal{L}_{\text{VD}}$	$\mathbf{FD_{incep}}\downarrow$	$\mathbf{FD_{dinov2}}\downarrow$	$\mathbf{ULIP_I}\uparrow$
×	×	20.26	195.6	34.62
✓	×	18.42	172.0	35.99
✓	✓	18.09	164.2	36.88

Table 3: Ablation studies on our proposed two loss functions.

Quantitative Comparison

Since methods for distilling 3D diffusion models remain scarce, FlashVDM (Lai et al. 2025) is one of the few. We also compare against non-distilled methods (LGM (Tang et al. 2024), 3DTopia-XL (Chen et al. 2025c), Ln3Diff (Lan et al. 2024)) and the teacher model TRELLIS (Xiang et al. 2024). Apart from LGM, all others are based on 3D diffusion. The results are shown in Table 1.

Non-diffusion-based Method. We compare our method with LGM (Tang et al. 2024), one of the most influential non-3D-diffusion-based 3D generation methods. Compared to LGM, our approach distills an end-to-end 3D diffusion model, which has stronger 3D awareness, improves both appearance and geometry quality of the generated 3D assets, and reduces inference time.

Non-distilled 3D Diffusion Methods. We further compare with non-distilled 3D diffusion methods, including 3DTopia-XL (Chen et al. 2025c), Ln3Diff (Lan et al. 2024), and the teacher model TRELLIS (Xiang et al. 2024). After distillation, our method reduces the two 3D flow transformers of TRELLIS from 25 steps each (50 steps total) to 1 and 1 steps (2 steps in total), achieving high-quality 3D generation in just 0.68s. We achieve comparable performance on appearance metrics (FD_{incep}, $\text{FD}_{\text{dinov2}}$) and the geometry metric (ULIP_{I}) compared with the teacher model when the number of inference steps is 4. Compared to Ln3Diff and 3DTopia-XL (Chen et al. 2025c), our method outperforms it in both appearance and geometry evaluations with fewer than 12× the inference steps and inference time.

Distilled 3D Diffusion Method. Compared to the recent distilled 3D-shape diffusion model FlashVDM (Lai et al.

Figure 5: Qualitative results of ablation studies on our proposed two loss functions.

2025), we evaluate only its geometry metric ($ULIP_I$), since it does not generate appearance. We use the officially released code of FlashVDM and evaluate it with the default configuration. Our method and FlashVDM complete inference in a similar number of steps. On the geometry metric $ULIP_I$, our method outperforms FlashVDM.

Qualitative Comparison

Qualitative Comparison with Other Methods. We present a visual comparison of the results in Fig. 2. The results show that our method can generate high-quality 3D assets with both geometry and appearance. Since FlashVDM (Lai et al. 2025) does not generate appearance for 3D assets, our comparison with FlashVDM mainly focuses on geometry. Compared to FlashVDM, our method generates geometry with more details and better consistency with the conditioned images. Compared with the teacher model TRELLIS (Xiang et al. 2024), our method generates comparable 3D assets.

Comparison with CM methods. We conduct qualitative and quantitative comparison with three CM methods: CM (Song et al. 2023), PCM (Wang et al. 2024), and sCM (Lu and Song 2024). The results are shown in Fig. 4 and Table 2. As the basic consistency model, CM performs the worst. Due to the multi-phase design, PCM learns more stably and achieves a better performance than CM. sCM utilizes a continuous format of CM, which eliminates the discretization errors, thereby performing better than the other two CM methods. Derived from a different objective, our method largely outperforms all the CM methods, with more complete and fine-grained geometric and visual details.

Ablation Study

Velocity Matching. The quantitative results of the ablation study on VM are in Table 3. It can be found that directly applying VM to finetune TRELLIS improves both the geometric quality and the visual quality. In Fig. 5, applying VM significantly reduces the unwanted extra geometry generated by the model, while partially restoring the missing and incomplete geometric structures. Moreover, compared to the generation result of the non-distilled model, the distilled model exhibits enhanced geometric fidelity, particularly in fine structures such as the shape of car windows and the overall body geometry.

Velocity Distillation. The quantitative results of the ablation study on VD are in Table 3. With the help of VM, VD is capable of learning the geometry better. In Fig. 5, VD further eliminates the unwanted extra geometry and addresses the remaining incomplete regions in the geometry.

Joint Optimization. In Table 3, joint optimization with VM and VD achieves the best performance, demonstrating the complementarity between the two loss functions. In Fig. 3, under the joint effect of the two loss functions, both geometry and visual quality are significantly improved compared to the TRELLIS baseline. Our method significantly addresses the remaining incomplete regions in the geometry and improves the details.

Conclusion

We present a novel framework MDT-dist for few-step flow distillation in 3D generation. By formulating a primary objective as modeling the transport from the marginal distribution to the data distribution, our approach provides a more direct solution to few-step generation, in contrast to consistency models that rely on consistency constraints on the adjacent time steps. To effectively optimize this objective, we introduce two optimization objectives: Velocity Matching (VM), which converts the optimization target from the transport level to the velocity level, enabling tractable and stable matching of velocity fields between the student and the teacher, and Velocity Distillation (VD), which converts the optimization target from the transport level to the distribution level, leveraging the learned velocity fields to perform probability density distillation. When applied to TRELLIS, our method reduces sampling steps from 25 to 1–2 while preserving high geometric and visual fidelity. Experiments demonstrate that our approach significantly outperforms existing consistency distillation techniques, and achieves new state-of-the-art performance in few-step 3D generation.

Limitations and Future Work. Our method still requires a large amount of conditional images and geometric data to conduct few-step distillation training. While images are relatively easy to collect, high-quality geometric data is much more scarce and expensive, making the few-step 3D generation distillation costly. A possible solution is to eliminate the dependence on geometric data and take only conditional images as input, which reduces the cost and further scales up the distillation by leveraging abundant online images.

Acknowledgments

This work was supported by the NSFC under Grant 62322604 and 62576207.

References

Chen, J.; Zhu, L.; Hu, Z.; Qian, S.; Chen, Y.; Wang, X.; and Lee, G. H. 2025a. MAR-3D: Progressive Masked Auto-regressor for High-Resolution 3D Generation. *arXiv preprint arXiv:2503.20519*.

Chen, R.; Zhang, J.; Liang, Y.; Luo, G.; Li, W.; Liu, J.; Li, X.; Long, X.; Feng, J.; and Tan, P. 2025b. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *CVPR*.

Chen, Z.; Tang, J.; Dong, Y.; Cao, Z.; Hong, F.; Lan, Y.; Wang, T.; Xie, H.; Wu, T.; Saito, S.; et al. 2025c. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *CVPR*.

Collins, J.; Goel, S.; Deng, K.; Luthra, A.; Xu, L.; Gundogdu, E.; Zhang, X.; Yago Vicente, T. F.; Dideriksen, T.; Arora, H.; Guillaumin, M.; and Malik, J. 2022. ABO: Dataset and Benchmarks for Real-World 3D Object Understanding. *CVPR*.

Deitke, M.; Schwenk, D.; Salvador, J.; Weihs, L.; Michel, O.; VanderBilt, E.; Schmidt, L.; Ehsani, K.; Kembhavi, A.; and Farhadi, A. 2023. Objaverse: A universe of annotated 3d objects. In *CVPR*.

Fu, H.; Jia, R.; Gao, L.; Gong, M.; Zhao, B.; Maybank, S.; and Tao, D. 2021. 3d-future: 3d furniture shape with texture. *IJCV*.

Gao, J.; Sun, Y.; Liu, Y.; Tang, Y.; Zeng, Y.; Qi, D.; Chen, K.; and Zhao, C. 2025. StyleShot: a Snapshot on any Style. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–15.

Geng, Z.; Deng, M.; Bai, X.; Kolter, J. Z.; and He, K. 2025. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*.

Gupta, A.; Xiong, W.; Nie, Y.; Jones, I.; and Oğuz, B. 2023. 3dgen: Triplane latent diffusion for textured mesh generation. *arXiv preprint arXiv:2303.05371*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *NeurIPS*.

Hong, Y.; Zhang, K.; Gu, J.; Bi, S.; Zhou, Y.; Liu, D.; Liu, F.; Sunkavalli, K.; Bui, T.; and Tan, H. 2023. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*.

Jain, A.; Mildenhall, B.; Barron, J. T.; Abbeel, P.; and Poole, B. 2022. Zero-shot text-guided object generation with dream fields. In *CVPR*.

Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4).

Khanna*, M.; Mao*, Y.; Jiang, H.; Haresh, S.; Shacklett, B.; Batra, D.; Clegg, A.; Undersander, E.; Chang, A. X.; and Savva, M. 2023. Habitat Synthetic Scenes Dataset (HSSD-200): An Analysis of 3D Scene Scale and Realism Tradeoffs for ObjectGoal Navigation. *arXiv preprint*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lai, Z.; Zhao, Y.; Zhao, Z.; Liu, H.; Wang, F.; Shi, H.; Yang, X.; Lin, Q.; Huang, J.; Liu, Y.; et al. 2025. Unleashing vecset diffusion model for fast shape generation. *arXiv preprint arXiv:2503.16302*.

Lan, Y.; Hong, F.; Yang, S.; Zhou, S.; Meng, X.; Dai, B.; Pan, X.; and Loy, C. C. 2024. LN3Diff: Scalable Latent Neural Fields Diffusion for Speedy 3D Generation. In *ECCV*.

Li, J.; Tan, H.; Zhang, K.; Xu, Z.; Luan, F.; Xu, Y.; Hong, Y.; Sunkavalli, K.; Shakhnarovich, G.; and Bi, S. 2023. Instant3D: Fast Text-to-3D with Sparse-View Generation and Large Reconstruction Model. *arXiv preprint arXiv:2311.06214*.

Li, W.; Liu, J.; Chen, R.; Liang, Y.; Chen, X.; Tan, P.; and Long, X. 2024. CraftsMan: High-fidelity Mesh Generation with 3D Native Generation and Interactive Geometry Refiner. *arXiv preprint arXiv:2405.14979*.

Li, W.; Zhang, X.; Sun, Z.; Qi, D.; Li, H.; Cheng, W.; Cai, W.; Wu, S.; Liu, J.; Wang, Z.; et al. 2025. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*.

Liang, Y.; Yang, X.; Lin, J.; Li, H.; Xu, X.; and Chen, Y. 2023. LucidDreamer: Towards High-Fidelity Text-to-3D Generation via Interval Score Matching. arXiv:2311.11284.

Lin, C.-H.; Gao, J.; Tang, L.; Takikawa, T.; Zeng, X.; Huang, X.; Kreis, K.; Fidler, S.; Liu, M.-Y.; and Lin, T.-Y. 2023. Magic3d: High-resolution text-to-3d content creation. In *CVPR*.

Lipman, Y.; Chen, R. T.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2022. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, R.; Wu, R.; Hoorick, B. V.; Tokmakov, P.; Zakharov, S.; and Vondrick, C. 2023. Zero-1-to-3: Zero-shot One Image to 3D Object. *arXiv preprint arXiv:2303.11328*.

Liu, X.; Gong, C.; and Liu, Q. 2022. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*.

Long, X.; Guo, Y.-C.; Lin, C.; Liu, Y.; Dou, Z.; Liu, L.; Ma, Y.; Zhang, S.-H.; Habermann, M.; Theobalt, C.; et al. 2023. Wonder3D: Single Image to 3D using Cross-Domain Diffusion. *arXiv preprint arXiv:2310.15008*.

Lu, C.; and Song, Y. 2024. Simplifying, stabilizing and scaling continuous-time consistency models. *arXiv preprint arXiv:2410.11081*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2020. NeRF: Representing

Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Poole, B.; Jain, A.; Barron, J. T.; and Mildenhall, B. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Raj, A.; Kaza, S.; Poole, B.; Niemeyer, M.; Ruiz, N.; Mildenhall, B.; Zada, S.; Aberman, K.; Rubinstein, M.; Barron, J.; et al. 2023. Dreambooth3d: Subject-driven text-to-3d generation. *arXiv preprint arXiv:2303.13508*.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*. PMLR.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.

Shen, Q.; Yi, X.; Wu, Z.; Zhou, P.; Zhang, H.; Yan, S.; and Wang, X. 2024. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*.

Shi, Y.; Wang, P.; Ye, J.; Long, M.; Li, K.; and Yang, X. 2023. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y.; Dhariwal, P.; Chen, M.; and Sutskever, I. 2023. Consistency Models. In *ICML*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2020. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Stojanov, S.; Thai, A.; and Rehg, J. M. 2021. Using shape to categorize: Low-shot learning with an explicit shape bias. In *CVPR*.

Sun, J.; Zhang, B.; Shao, R.; Wang, L.; Liu, W.; Xie, Z.; and Liu, Y. 2023. DreamCraft3D: Hierarchical 3D Generation with Bootstrapped Diffusion Prior. *arXiv preprint arXiv:2310.16818*.

Tang, J.; Chen, Z.; Chen, X.; Wang, T.; Zeng, G.; and Liu, Z. 2024. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation. *arXiv preprint arXiv:2402.05054*.

Tang, J.; Ren, J.; Zhou, H.; Liu, Z.; and Zeng, G. 2023. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv preprint arXiv:2309.16653*.

Tong, A.; Malkin, N.; Huguet, G.; Zhang, Y.; Rector-Brooks, J.; Fatras, K.; Wolf, G.; and Bengio, Y. 2023. Conditional flow matching: Simulation-free dynamic optimal transport. *arXiv preprint arXiv:2302.00482*, 2(3).

Wang, F.-Y.; Huang, Z.; Bergman, A.; Shen, D.; Gao, P.; Lingelbach, M.; Sun, K.; Bian, W.; Song, G.; Liu, Y.; et al. 2024. Phased consistency models. *NeurIPS*.

Wang, X.; Liu, L.; Cao, Y.; Wu, R.; Qin, W.; Wang, D.; Sui, W.; and Su, Z. 2025. EmbodiedGen: Towards a Generative 3D World Engine for Embodied Intelligence. arXiv:2506.10600.

Wang, Z.; Lu, C.; Wang, Y.; Bao, F.; Li, C.; Su, H.; and Zhu, J. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. *arXiv preprint arXiv:2305.16213*.

Wu, S.; Lin, Y.; Zhang, F.; Zeng, Y.; Xu, J.; Torr, P.; Cao, X.; and Yao, Y. 2024. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. *arXiv preprint arXiv:2405.14832*.

Wu, S.; Lin, Y.; Zhang, F.; Zeng, Y.; Yang, Y.; Bao, Y.; Qian, J.; Zhu, S.; Torr, P.; Cao, X.; and Yao, Y. 2025. Direct3D-S2: Gigascale 3D Generation Made Easy with Spatial Sparse Attention. *arXiv preprint arXiv:2505.17412*.

Xiang, J.; Lv, Z.; Xu, S.; Deng, Y.; Wang, R.; Zhang, B.; Chen, D.; Tong, X.; and Yang, J. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506*.

Xu, Y.; Shi, Z.; Yifan, W.; Peng, S.; Yang, C.; Shen, Y.; and Gordon, W. 2024. GRM: Large Gaussian Reconstruction Model for Efficient 3D Reconstruction and Generation. *arxiv: 2403.14621*.

Yang, X.; Shi, H.; Zhang, B.; Yang, F.; Wang, J.; Zhao, H.; Liu, X.; Wang, X.; Lin, Q.; Yu, J.; et al. 2024. Hunyuan3D 1.0: A Unified Framework for Text-to-3D and Image-to-3D Generation. *arXiv preprint arXiv:2411.02293*.

Ye, C.; Wu, Y.; Lu, Z.; Chang, J.; Guo, X.; Zhou, J.; Zhao, H.; and Han, X. 2025. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. *arXiv preprint arXiv:2503.22236*.

Yi, T.; Fang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Tian, Q.; and Wang, X. 2023. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arXiv preprint arXiv:2310.08529*.

Yi, T.; Fang, J.; Zhou, Z.; Wang, J.; Wu, G.; Xie, L.; Zhang, X.; Liu, W.; Wang, X.; and Tian, Q. 2024. Gaussiandreamer-pro: Text to manipulable 3d gaussians with highly enhanced quality. *arXiv preprint arXiv:2406.18462*.

Zhang, K.; Bi, S.; Tan, H.; Xiangli, Y.; Zhao, N.; Sunkavalli, K.; and Xu, Z. 2024a. GS-LRM: Large Reconstruction Model for 3D Gaussian Splatting. *ECCV*.

Zhang, L.; Wang, Z.; Zhang, Q.; Qiu, Q.; Pang, A.; Jiang, H.; Yang, W.; Xu, L.; and Yu, J. 2024b. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)*.

Zhao, M.; Zhao, C.; Liang, X.; Li, L.; Zhao, Z.; Hu, Z.; Fan, C.; and Yu, X. 2023. EfficientDreamer: High-Fidelity and Robust 3D Creation via Orthogonal-view Diffusion Prior. *arXiv preprint arXiv:2308.13223*.

Zhao, Z.; Lai, Z.; Lin, Q.; Zhao, Y.; Liu, H.; Yang, S.; Feng, Y.; Yang, M.; Zhang, S.; Yang, X.; et al. 2025. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*.