Point-MF: Stabilizing One-Step Mean Flows for Single-Image Point Cloud Reconstruction

Yuta Baba[1] and Keiji Yanai[1]

The University of Electro-Communications, Tokyo, Japan
baba-y@mm.inf.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. Single-image point cloud reconstruction requires recovering complete object-level geometry, including occluded regions, from a single RGB image. Diffusion- and flow-based reconstructors can model this ambiguity, but their iterative sampling requires many network function evaluations, while naive one-step point-space updates often produce outliers and density imbalance. We propose Point-MF, a one-step conditional Mean-Flow framework for geometry-only point cloud reconstruction. Point-MF predicts an interval-averaged velocity field directly in point-cloud space and reconstructs the output with a single network function evaluation, without training a point-cloud autoencoder. To stabilize large Mean-Flow jumps, we introduce Denoised Space Anchor (DSA), a set-distance auxiliary loss that anchors the denoised point set implied by the predicted velocity to the ground-truth geometry. On all 13 ShapeNet-R2N2 categories and Pix3D, Point-MF achieves strong reconstruction quality under the aligned point-set protocol, improving average CD and EMD over the evaluated point-cloud reconstruction baselines. It runs in 40.72 ms per sample on an RTX A4000, within the same latency order as the feedforward RGB2point baseline and substantially faster than iterative diffusion baselines such as PC^2 and BDM. These results show that denoised-space geometric anchoring enables stable one-step Mean Flow directly in point-cloud space.

Keywords: Point cloud reconstruction · Single-image 3D reconstruction · Mean Flow · Few-step generation

1 Introduction

Single-image point cloud reconstruction is an inherently ill-posed problem: a model must infer a complete 3D shape, including self-occluded and unobserved regions, from a single RGB observation. Since the same visible evidence can correspond to multiple plausible 3D completions, this task requires both image-conditioned shape inference and a mechanism for handling ambiguity. Fast reconstruction of sparse object-level geometry is useful for applications such as spatial understanding, robotic perception, AR/VR preprocessing, and lightweight downstream 3D analysis, where latency and memory cost are important practical constraints.

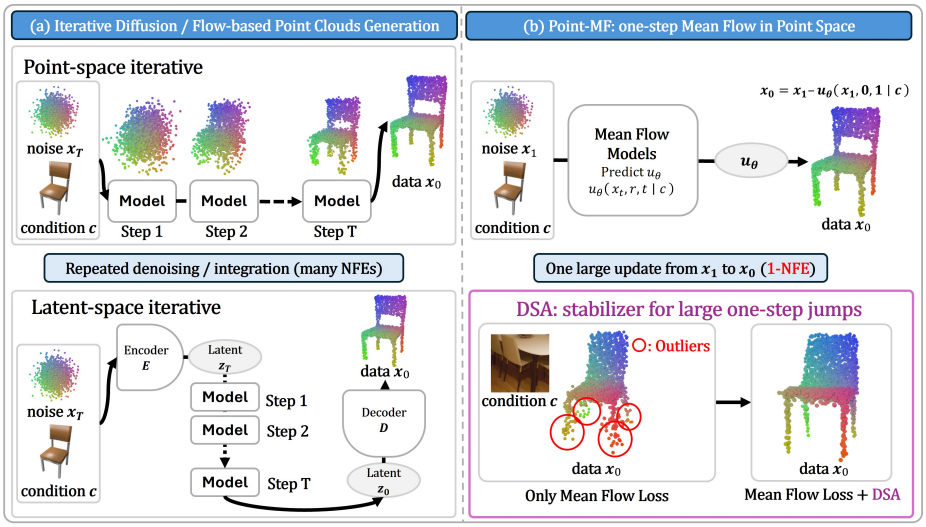


Fig. 1: Conceptual comparison between Point-MF and iterative point-cloud generation. (a) Many existing diffusion/flow-based point-cloud generators obtain reconstructions through iterative denoising or numerical integration, requiring many network function evaluations (NFEs). (b) Point-MF directly predicts an interval-averaged velocity field in point-cloud space and performs a one-step update from noise to the reconstructed point cloud. DSA regularizes the induced data-side estimate with a set-distance loss, suppressing outliers and density artifacts caused by one-step updates.

Point clouds support set-distance supervision and can serve as intermediate representations for surface reconstruction or rendering [3, 20, 26, 34, 68], or as inputs to grasp prediction [11]. When only point geometry is needed, direct prediction avoids reconstructing mesh connectivity that would be discarded after point sampling. We therefore target fast, aligned geometry-only reconstruction rather than textured asset generation.

Existing single-image 3D methods can be broadly grouped into direct regression methods, probabilistic diffusion/flow-based generative methods, and open-category image-to-3D models. Direct regression methods [2, 9, 19, 21, 35, 63, 64] predict a 3D representation from an input image in a single forward pass and are therefore attractive for fast inference. Among point-cloud reconstruction methods, RGB2point [35] is a strong practical baseline. However, deterministic prediction makes it difficult to represent multiple plausible shape completions for occluded regions.

In contrast, diffusion/flow-based methods [5, 8, 22, 27, 33, 43, 45, 49, 67, 70] can probabilistically model shape ambiguity, but they typically require many NFEs due to iterative denoising or numerical integration. More recently, open-category image-to-3D models and asset generators, including the TRELLIS series [65, 66] and related methods [2, 17, 20, 25, 32], have substantially advanced high-resolution 3D generation. These models are important reference points, but many of them target 3D asset generation with meshes, structured 3D latent representations,

3D Gaussians, or appearance and material information. This differs from the aligned geometry-only point cloud reconstruction setting studied in this work.

Mean Flow [13], derived from Flow Matching [38], enables large updates with few sampling steps by learning an interval-averaged velocity field between two time points. However, directly applying this idea to unordered point-cloud space is non-trivial. An N-point cloud lies in a high-dimensional set space, and even small velocity errors in a one-step update can appear as outliers, density imbalance, or deviations from the underlying shape manifold. Therefore, the challenge is to reduce NFEs while stabilizing the clean geometry (a data-side estimate at $t = 0$) induced by a large Mean Flow update.

To address this challenge, we propose **Point-MF**, a one-step conditional Mean Flow framework for single-image point cloud reconstruction (Fig. 1). Point-MF predicts the mean velocity field directly in point-cloud space without using a point-cloud autoencoder, and reconstructs the output with a single network function evaluation. The central component is **Denoised Space Anchor** (DSA). DSA anchors the clean point-cloud estimate x_θ induced by the predicted velocity to the ground-truth geometry using a set-distance loss, thereby suppressing outliers and density artifacts caused by large one-step updates.

We evaluate Point-MF on ShapeNet-R2N2 and Pix3D in terms of both shape quality and efficiency. On ShapeNet-R2N2, we report results on all 13 categories and compare Point-MF with feedforward methods and iterative diffusion-based methods. For recent image-to-3D models, we provide representative qualitative comparisons in the main paper, and report quantitative reference evaluations involving point-cloud conversion in the supplementary material.

Our contributions are summarized as follows:

- We formulate single-image point cloud reconstruction as one-step conditional Mean Flow directly in point-cloud space, and propose Point-MF, a geometry-only reconstruction framework that does not require a point-cloud autoencoder.
- We introduce Denoised Space Anchor (DSA), which regularizes the clean point-cloud estimate induced by the predicted mean velocity using a set-distance loss, reducing outliers and density artifacts caused by one-step updates.
- We validate Point-MF through comprehensive experiments on all 13 ShapeNet-R2N2 categories and Pix3D, including reconstruction quality, inference efficiency, memory usage, and ablation studies.

2 Related Work

3D Reconstruction from Images. Image-based 3D reconstruction has explored a wide range of 3D representations, including voxels, meshes, point clouds, implicit functions, and Gaussian Splatting [9, 29, 44, 46, 47, 50, 62, 63]. For single-image reconstruction, prior work has proposed voxel-based methods such as 3D-R2N2 [9], mesh-based methods such as Pixel2Mesh and AtlasNet [16, 63], and implicit-function-based methods such as Occupancy Networks and DeepSDF [44,

50]. In contrast, point clouds provide a lightweight geometry representation and can be directly supervised with set distances, making them suitable for geometry-only single-image reconstruction. Representative point-cloud reconstruction methods include reprojection-based approaches [24] and RGB2point [35], which performs fast direct point-cloud generation.

More recently, probabilistic generation of point clouds and 3D shapes has been widely studied. Point cloud diffusion [42], DiT-3D [45], Point-E [49], PC^2 [43], and BDM [67] are representative examples that extend diffusion models to 3D generation or single-image reconstruction. These methods can probabilistically model ambiguous 3D completion, but they require many iterative updates at inference time, resulting in large NFEs. In contrast, LION [70] is a representative latent-space diffusion method that uses a hierarchical point-cloud VAE and performs diffusion generation in a latent space consisting of a global shape latent and a point-structured latent. While LION achieves stronger generation performance than DDMs operating directly in point-cloud space, it depends on the training and reconstruction quality of a point-cloud autoencoder and still requires iterative sampling during generation. In contrast, our method learns one-step conditional Mean Flow directly in point-cloud space without using a point-cloud autoencoder.

In addition, open-category image-to-3D models and 3D asset generation methods, such as LRM [19], the TRELLIS series [65,66], SPAR3D [20], SAM3D [6], ZeroShape/ZeroShape-W [7, 23], and SparseFlex [17], have made substantial progress in high-resolution and open-category 3D generation. However, many of these methods target 3D asset generation with meshes, structured 3D latent representations, dense geometry, or appearance information. This differs from the aligned geometry-only point cloud reconstruction setting studied in this work in terms of output representation, training data, and evaluation protocol. Therefore, we treat these methods not as primary in-domain competitors, but as reference methods under a point-sampled geometry evaluation protocol.

Diffusion and Flow-based Models. Diffusion and flow-based generative models learn a mapping from a prior distribution to a data distribution as a continuous-time generative trajectory [18, 28, 38, 39, 56, 59, 60]. While these methods enable high-quality generation, they often require iterative numerical integration at inference time, which leads to high computational cost. Recent work has also studied regularizing the clean output x_0 estimated from intermediate states along the generative trajectory to improve few-step generation quality [1, 37, 57, 58]. In this work, we extend this clean-output supervision principle to point clouds by anchoring the induced denoised point set to the ground-truth geometry using a set-distance loss.

Few-step Generative Models. Standard diffusion and flow-based models assume multi-step numerical integration, whereas recent methods aim at one-step or few-step generation by directly learning large temporal updates [12, 13, 15, 30, 40, 57, 58, 72]. Consistency Models, CTM, Shortcut Models, and IMM accelerate generation through direct mappings to terminal samples, trajectory learning

between arbitrary time pairs, or moment matching [12, 30, 72]. Mean Flow [13] enables low-NFE generation by learning the mean velocity field between two time points, and several follow-up methods have improved its stability and sampling behavior [14, 36, 71]. However, as shown by Pixel Mean Flows [41], naive u-prediction can become unstable in high-dimensional latent-free spaces. A similar issue arises in point-cloud space: errors in a one-step velocity prediction can manifest as outliers, density imbalance, or shape deviations. We therefore stabilize the induced denoised estimate geometrically using DSA.

3 Preliminaries

Flow Matching. Flow Matching (FM) [38] learns a velocity field along a transport path between a data distribution and a prior distribution. Given a data sample $x \sim p_{\text{data}}$ and noise $\epsilon \sim p_{\text{prior}}$, we use the linear interpolation $x_t = (1-t)x+t\epsilon$, whose path-wise velocity is $\epsilon - x$. FM regresses the marginal velocity field induced by this path, $v(x_t, t) = \mathbb{E}[\epsilon - x \mid x_t]$. Under our time convention, $t = 0$ corresponds to the data side and $t = 1$ to the noise side. At inference time, samples are generated by integrating the learned ODE backward from $t = 1$ to $t = 0$.

Mean Flow. Mean Flow (MF) [13] enables low-NFE generation by directly learning an interval-averaged velocity field between two time points, rather than the instantaneous velocity field used in Flow Matching. Under our time convention, $t = 0$ corresponds to the data side and $t = 1$ to the noise side. For $0 \le r < t \le 1$, the mean velocity field for moving from the state x_t at time t to the state x_r at time r is defined as

$$u^\star(x_t, r, t) \coloneqq \frac{1}{t-r} \int_r^t v^\star(x_\tau, \tau)\, d\tau. \tag{1}$$

Using this mean velocity field, the update over the interval $[r, t]$ can be written as

$$x_r = x_t - (t-r)u^\star(x_t, r, t). \tag{2}$$

Therefore, if $u^\star$ can be directly approximated by a neural network, a large temporal jump can be performed without many numerical integration steps.

In practice, instead of directly evaluating the integral in Eq. (1), MF uses a self-consistent regression target derived from the Mean Flow identity. In this work, we introduce an image condition c and train a conditional MF model $u_\theta(x_t, r, t \mid c)$ with classifier-free condition dropout. The detailed derivation of Classifier-Free Guidance (CFG) -guided Mean Flow and the target computation using Jacobian–vector product (JVP) are provided in Supplementary Sec. A. In the main paper, we use the following objective:

$$\mathcal{L}_{\text{MF-CFG}}(\theta) = \mathbb{E}\left[\left\| u_\theta(x_t, r, t \mid c) - \text{sg}\left(u_{\text{tgt}}^{\text{cfg}}\right)\right\|_2^2\right], \tag{3}$$

where $u_{\text{tgt}}^{\text{cfg}}$ denotes the CFG-guided regression target based on the Mean Flow identity, and $\text{sg}(\cdot)$ denotes stop-gradient.

At inference time, we fix $(r, t) = (0, 1)$ and generate a point cloud from noise $x_1 = \epsilon$ by

$$x_0 = \epsilon - u_\theta(\epsilon, 0, 1 \mid c), \tag{4}$$

using a single network function evaluation.

4 Method

This section describes Point-MF, a conditional Mean Flow framework for single-image point cloud reconstruction. We first describe the network architecture, focusing on image conditioning, interval/time conditioning, and the proposed Condition-Token Integrator (CTI). An overview is shown in Fig. 2. We then present the loss design for stabilizing one-step Mean Flow generation and improving geometric consistency in data space.

4.1 Architecture

Image Conditioning with DINOv3. In single-image point cloud reconstruction, the model must infer complete 3D geometry from partial visual evidence. We use a frozen DINOv3 encoder [55] to extract image features from the input image. The encoder outputs a global image feature $\mathbf{z}_{\text{img}} \in \mathbb{R}^D$ corresponding to the class token, and patch-sequence features $Z_{\text{ctx}} \in \mathbb{R}^{M \times D}$. The global feature is projected to the hidden dimension of the DiT and used as $e_{\text{img}} \in \mathbb{R}^H$ for global modulation. The patch features are fed into the Condition-Token Integrator (CTI), described in Sec. 4.1, and are transformed into layer-wise context tokens for Cross-Attention. The DINOv3 encoder is frozen, while the projection layers, CTI, and point-cloud generator are trained for the reconstruction objective.

Mean-Flow-Compatible Point-Cloud DiT. The point-cloud generator is implemented as a Transformer composed of L Diffusion Transformer (DiT) blocks. Since a point cloud is an unordered set, we follow prior point-cloud diffusion models [49] and do not use point-index-dependent positional embeddings, thereby preserving permutation equivariance with respect to the input point order. The DiT backbone is based on the sparse point-cloud generation Transformer of Lan *et al.* [34]. Unlike standard diffusion models, our network predicts the interval-averaged velocity field $u_\theta(x_t, r, t \mid c)$ rather than an instantaneous velocity. Therefore, in addition to the current time t, we condition the model on the interval length $dt = t - r$.

Following DiT [51], we map t and dt into $e_t(t), e_{dt}(dt) \in \mathbb{R}^H$ using sinusoidal embeddings followed by two-layer MLPs. We then add the global image embedding e_{img} to obtain the conditioning vector $\mathbf{I} = e_t(t) + e_{dt}(dt) + e_{\text{img}}$. This vector is used to generate the AdaLN-Zero modulation parameters in each DiT block.

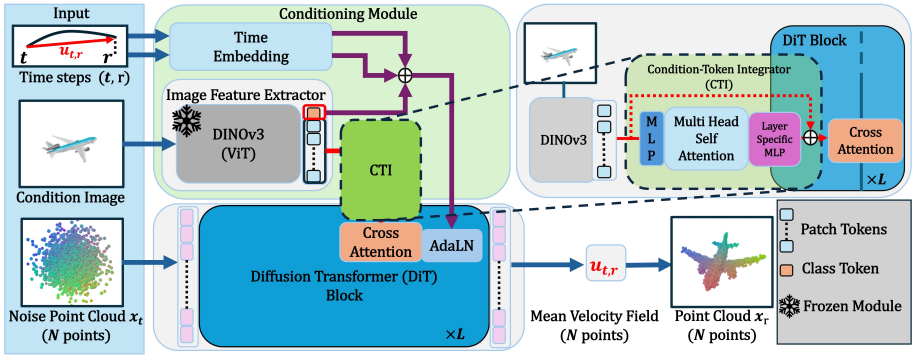


Fig. 2: Overview of Point-MF. Global features from a frozen DINOv3 encoder are combined with the embeddings of time t and interval length $dt = t - r$, and are used for AdaLN-Zero modulation. In CTI, patch tokens are integrated once by shared processing, and each DiT block forms its Cross-Attention context through a layer-specific MLP. The point-cloud DiT predicts the mean velocity field $u_\theta(x_t, r, t \mid c)$ from the noisy point cloud x_t, and generates the point cloud through the Mean Flow update $x_r = x_t - (t - r)u_\theta(x_t, r, t \mid c)$.

Since Mean Flow estimates an average velocity over an interval rather than an instantaneous velocity at time t, explicitly conditioning on dt enables the model to handle different temporal jumps. The detailed block-level architecture is provided in the Supplementary Sec. B.

Condition-Token Integrator (CTI). Patch tokens from a frozen DINOv3 encoder contain strong semantic information, but they are not directly optimized as Cross-Attention conditions for point-cloud generation. We therefore introduce the *Condition-Token Integrator (CTI)*, which transforms image patch tokens into layer-wise context tokens for the point-cloud DiT. CTI consists of a token integration stage applied once on the condition branch and a lightweight layer-specific injection stage inside each DiT block.

Let $Z_{\text{ctx}} \in \mathbb{R}^{M \times D}$ denote the patch features extracted by DINOv3. We first project Z_{ctx} with an input projection ϕ_{in} to obtain $Z_{\text{base}} = \phi_{\text{in}}(Z_{\text{ctx}}) \in \mathbb{R}^{M \times H}$. We then apply MHSA once to the projected image tokens and obtain integrated features $A_{\text{CTI}} = \text{MHSA}(\phi_{\text{MHSA}}(Z_{\text{base}}))$. In the ℓ-th DiT block, this integrated feature is projected by a layer-specific MLP ψ_ℓ and added to the base image tokens:

$$Z_{\text{CTI}}^{(\ell)} = Z_{\text{base}} + \psi_\ell(A_{\text{CTI}}). \tag{5}$$

The resulting $Z_{\text{CTI}}^{(\ell)}$ is used as the Cross-Attention context in the ℓ-th DiT block.

In this design, token interactions are computed only once on the condition branch, while each DiT block applies only a lightweight layer-specific projection. Thus, CTI provides a task-specific interface between frozen DINOv3 features and the point-cloud DiT while keeping the generator compact.

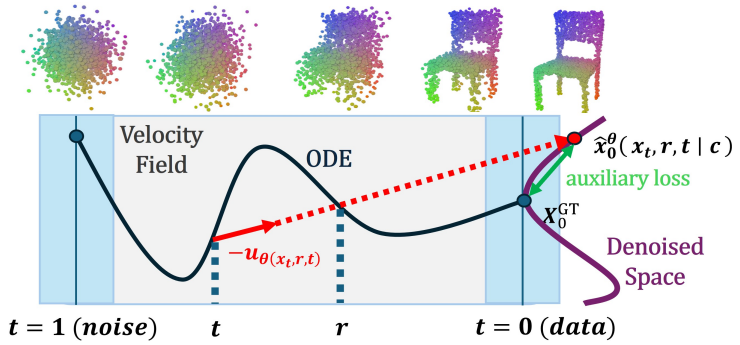


Fig. 3: Conceptual illustration of Denoised Space Anchor (DSA). Point-MF extrapolates the predicted mean velocity field $u_\theta(x_t, r, t \mid c)$ toward the data time $t = 0$ to obtain a denoised-space estimate $\hat{x}_0^\theta$. DSA anchors this estimated point cloud to the ground-truth point cloud x_0^{GT} using a set-distance loss, thereby suppressing outliers and density imbalance caused by one-step Mean Flow updates.

4.2 Training Objective

Denoised Space Anchor. Mean Flow enables low-NFE generation by predicting an interval-averaged velocity field for a large temporal jump. In point-cloud space, however, small velocity errors can directly lead to outliers, surface misalignment, or density imbalance in the reconstructed shape. This is particularly problematic in one-step generation, where there are no later correction steps.

To address this issue, we introduce *Denoised Space Anchor (DSA; Fig. 3)*, an auxiliary loss that regularizes the clean point cloud implied by the predicted mean velocity. Given a sampled state x_t and a predicted mean velocity $u_\theta(x_t, r, t \mid c)$, we form a denoised-space estimate by extrapolating the prediction to the clean endpoint:

$$\hat{x}_0^\theta(x_t, r, t \mid c) = x_t - t\, u_\theta(x_t, r, t \mid c). \tag{6}$$

When $r = 0$, Eq. (6) has the Mean-Flow update form. For general (r, t), extrapolation to $t = 0$ is exact for the constant velocity of a sample-wise linear path, but not generally for the marginal field [13]; we use it as an auxiliary geometric estimate.

Since point clouds are unordered sets, a point-wise loss is not appropriate unless a correspondence is given. We therefore use Adaptive Probabilistic Matching Loss (APML) [53] as the set distance:

$$\mathcal{L}_{\mathrm{DSA}} = \mathcal{L}_{\mathrm{APML}}\left(\hat{x}_0^\theta(x_t, r, t \mid c), x_0^{\mathrm{GT}}\right). \tag{7}$$

APML provides soft matching between the predicted and ground-truth point sets, making it better suited to the denoised-space estimate than a direct point-wise regression loss. In Sec. 5.1, we further compare APML with Chamfer Distance and MSE as the set distance used in DSA.

DSA is used as an auxiliary constraint and does not replace the Mean-Flow velocity supervision. It instead pulls the clean point cloud induced by the

predicted velocity field toward geometrically plausible shapes. To balance the strength of this constraint across time, we use the time-dependent weight

$$\lambda(t) = \mathbf{1}_{t \neq r} \frac{\lambda_{\text{base}}}{\max(t, \tau)}. \tag{8}$$

We set $\tau = 10^{-5}$ to bound the weight near $t = 0$. The factor $1/\max(t, \tau)$ compensates for the factor t in Eq. (6), while $\mathbf{1}_{t \neq r}$ disables DSA for the local Flow-Matching-like branch ($t = r$).

Finally, to absorb the scale difference between the Mean-Flow objective and the set distance, we use a mini-batch-based scaling coefficient:

$$s = \frac{\mathbb{E}[\mathcal{L}_{\text{MF-CFG}}]_{\text{detach}}}{\mathbb{E}[\mathcal{L}_{\text{DSA}}]_{\text{detach}} + \delta}. \tag{9}$$

The stop-gradient on the batch averages prevents the scaling coefficient itself from becoming an optimization target. The small constant δ is used only for numerical stability; we set $\delta = 10^{-8}$ in all experiments.

The final training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MF-CFG}} + s\, \lambda(t)\, \mathcal{L}_{\text{DSA}}. \tag{10}$$

For $t \neq r$, the explicit DSA gradient-scale factor is $s\lambda_{\text{base}} \min(1, t/\tau)$, so the clipping affects this factor only for $t < \tau$. In our experiments, we set $\lambda_{\text{base}} = 0.5$, which provides the best EMD-oriented trade-off in the DSA weight ablation.

5 Experiments

We evaluate Point-MF in terms of both reconstruction quality and inference efficiency. Since our goal is to generate geometrically consistent point clouds with low-NFE sampling, we report not only CD, EMD, and F-Score, but also inference time and peak memory usage.

Datasets. We evaluate Point-MF on ShapeNet-R2N2 [4] and Pix3D [61]. For ShapeNet-R2N2, we follow the 13-category split used in prior work [9,48], sample target point clouds from each 3D model, and normalize them to be centered at the origin. For Pix3D, we follow the split of Xu *et al.* [67]. Pix3D is used only as a test-only benchmark to evaluate real-image generalization, and is not used for training.

Metrics and Evaluation Protocol. We evaluate the reconstruction quality between the generated point cloud $\hat{X}$ and the ground-truth point cloud X using L1 Chamfer Distance (CD), Earth Mover's Distance (EMD), and F-Score. CD and EMD are used as complementary metrics for measuring geometric accuracy and global point-correspondence consistency. For CD and EMD, we unify the output resolution to 1024 points on ShapeNet-R2N2 and 2048 points on Pix3D. F-Score is computed with a threshold of 1% of the longest side of the bounding

Table 1: Comparison of CD and EMD on ShapeNet-R2N2. NFE denotes inference-time network evaluations; BDM uses 256+20 = 276 NFEs including its prior model. "–" denotes unavailable or incompatible comparisons. ViT-B uses a standard ImageNet-pretrained encoder [10]. For FCM [54], we report both 1-view and 2-view settings.

		CD×100 ↓				EMD×100 ↓			
Method	NFE	Car	Chair	Air	Avg.	Car	Chair	Air	Avg.
Self-Sup. [48]	1	5.48	10.91	7.11	7.85	4.95	14.93	11.07	9.99
DIFFER [31]	1	6.35	9.78	5.67	7.47	6.03	16.21	9.90	10.65
ULSP [24]	1	5.40	9.72	5.91	7.11	4.78	10.18	7.66	7.41
RGB2point [35]	1	**4.22**	5.43	**2.70**	4.33	5.63	9.53	6.86	7.34
PC^2 [43]	256	4.95	5.60	3.37	4.84	4.69	5.36	4.27	4.85
BDM [67]	276	5.12	5.02	3.12	4.64	4.25	4.53	3.92	4.28
LION [70]	1000	5.86	12.53	5.95	8.35	6.54	13.73	8.08	9.54
FCM (1-View) [54]	256	4.93	–	3.52	–	5.57	–	5.39	–
FCM (2-View) [54]	256	4.64	–	2.91	–	5.51	–	5.17	–
Ours	1	**4.22**	**4.29**	2.76	**3.92**	**3.82**	**4.29**	**3.04**	**3.82**
w/o CTI	1	4.58	4.29	3.01	4.13	4.02	4.62	3.54	4.14
w/o DSA	1	5.21	6.16	3.71	5.23	4.89	6.08	4.12	5.16
DINOv3 → ViT	1	4.12	4.45	2.72	3.93	4.02	4.31	3.12	3.93

box. Avg. is the instance-weighted mean over listed categories (ShapeNet-R2N2 counts: Supplementary Table 3; Pix3D images: chair/sofa/table = 768/390/374). We reimplemented RGB2point.

For F-Score, checkmarks in Table 2 indicate RepKPU [52] upsampling from 2048 to 8192 points; unmarked Point-MF results use direct 8192-point generation without retraining. This is enabled by the absence of index-dependent positional embeddings and does not imply resolution-invariant performance. Additional qualitative comparisons and complete 13-category quantitative results, including image-to-3D reference comparisons, are provided in Supplementary Secs. E and F, respectively.

To evaluate the practical efficiency of low-NFE generation, we compare inference time and peak memory usage. GPU inference is measured on a single NVIDIA RTX A4000. After warm-up, we run each method 100 times on the same input and report the mean and standard deviation. We compare against PC^2 [43], BDM [67], and RGB2point [35], with the output resolution fixed to 2048 points. Furthermore, to examine the feasibility of lightweight deployment, we additionally evaluate Core ML inference on an iPhone 15 Pro as a supplementary experiment.

Table 2: F-Score comparison on ShapeNet-R2N2 with a threshold of 1%. ✓ indicates evaluation after RepKPU upsampling from 2048 to 8192 points.

Category	LegoFormer [69]	PC^2 [43]	BDM [67]	LION [70]	RGB2point [35]	Ours	Ours
Upsample		✓	✓	–	–	–	✓
airplane	0.215	0.473	0.512	0.560	0.583	0.591	**0.603**
car	0.220	0.301	0.311	0.272	0.339	0.338	**0.342**
chair	0.217	0.202	0.238	0.145	0.195	0.243	**0.252**
sofa	**0.260**	0.177	0.211	–	0.194	0.238	0.240
table	0.305	0.238	0.268	–	0.268	0.309	**0.316**
Average	0.247	0.271	0.299	–	0.304	0.332	**0.339**

Table 3: Comparison of CD and EMD on the Pix3D dataset.

		CD×100 ↓				EMD×100 ↓			
Method	NFE	Chair	Sofa	Table	Avg.	Chair	Sofa	Table	Avg.
ZeroShape [23]	1	8.66	7.12	8.40	8.20	9.12	7.54	10.10	8.96
PC^2 [43]	256	7.07	5.87	11.39	7.82	7.13	6.14	10.72	7.75
BDM [67]	276	6.62	5.55	8.35	6.77	7.05	**4.98**	8.79	6.95
RGB2point [35]	1	6.62	5.40	9.70	7.06	9.34	7.11	12.69	9.59
Ours	1	**6.48**	**5.02**	**8.20**	**6.53**	**6.74**	5.06	**7.94**	**6.61**
w/o CTI	1	7.58	5.58	8.45	7.28	7.23	5.52	8.34	7.07
w/o DSA	1	8.82	6.58	10.12	8.57	8.88	7.02	9.76	8.62
DINOv3 → ViT	1	6.58	5.32	8.11	6.63	6.84	5.12	7.94	6.67

Implementation Details. All evaluated Point-MF models are trained in PyTorch on eight NVIDIA RTX A6000 GPUs (∼25 hours); a separate JAX/XLA implementation takes ∼16 hours. Each training sample consists of an $N = 2048$ point cloud and a 224×224 RGB image. We use pretraining only for the image feature extractor, namely DINOv3, and train all other modules from scratch. The DiT backbone has hidden dimension $H = 512$, $L = 12$ blocks, and $h = 8$ attention heads. For image conditioning, we use DINOv3 ViT-B as the frozen image encoder. The CTI module for processing DINO patch tokens is applied once on the condition branch, with internal dimension 1024 and 4 attention heads.

The DSA weight is set to $\lambda_{\text{base}} = 0.5$. We optimize the model using AdamW with learning rate 1.0×10^{-4}, batch size 128, and 120,000 training steps. The first 10,000 steps are used for warm-up. The CFG-based velocity composition used during training is included in the objective in Eq. (3). Additional hyperparameter details are provided in Supplementary Sec. B.

Quantitative Results. Table 1 reports CD and EMD on ShapeNet-R2N2, and Table 2 compares F-Score. Results on Pix3D are shown in Table 3. As shown in Table 1, Point-MF outperforms diffusion-based methods that require multi-step sampling, while using only one NFE.

Table 4: Inference time and peak memory usage. GPU results are measured on an RTX A4000, and mobile results are measured on an iPhone 15 Pro using Core ML.

Method	NFE	Time [ms/sample]	Peak Mem.
Device: RTX A4000			
PC^2 [43]	256	22000 ± 200	1.730 GiB
BDM [67]	276	28000 ± 200	1.970 GiB
RGB2point [35]	1	28.39 ± 2.18	0.818 GiB
Ours (2048 points)	1	40.72 ± 2.74	1.033 GiB
Ours (8192 points)	1	84.61 ± 1.21	1.097 GiB
Device: iPhone 15 Pro (Core ML)			
Ours (2048 points)	1	281.74 ± 1.51	0.608 GiB

RGB2point is trained with a CD objective, and therefore has an advantage on CD because its training objective is aligned with the evaluation metric.

However, compared with probabilistic generative methods such as PC^2, BDM, and Point-MF, RGB2point performs worse in terms of EMD. As shown in Table 2, Point-MF achieves the best average F-Score among the compared methods. Notably, Point-MF retains strong performance when directly generating 8192 points without additional training or RepKPU upsampling. On the real-image Pix3D dataset, Point-MF also achieves the best overall performance with only one NFE, as shown in Table 3. Table 4 compares inference time and peak

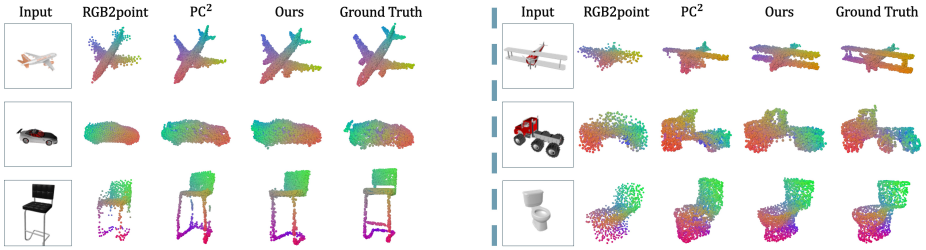


Fig. 4: Qualitative results on ShapeNet.

memory usage. Point-MF is 1.43× slower than RGB2point (40.72 vs. 28.39 ms), but over 500× faster than PC^2 and BDM under the same RTX A4000 protocol. Runtime increases when directly generating 8192 points due to the larger point-token count. We also convert Point-MF to Core ML and verify its execution on an iPhone 15 Pro. This mobile evaluation is not intended as a same-environment comparison with existing methods, but rather as a supplementary experiment showing that the 1-NFE inference of Point-MF is feasible on a real device. Details of the Core ML conversion and mobile measurement protocol are provided in Supplementary Sec. C.

Qualitative Results. To visually evaluate shape fidelity and point-density distribution, we conduct qualitative comparisons on ShapeNet-R2N2 (Fig. 4) and Pix3D (Fig. 5). Overall, Point-MF generates point distributions with consistent shapes while suppressing density imbalance. This trend is consistent with the strong EMD performance of Point-MF in Table 1. In contrast, RGB2point, which directly minimizes Chamfer Distance, often produces sharp contours because points are strongly attracted to the surface, but it can suffer from local density imbalance or point concentration in specific regions. On Pix3D, Point-MF can also reconstruct challenging structures such as table undersides and complex chair legs, which are often missed by PC^2 and RGB2point. We further show a subset of qualitative comparisons with recent image-to-3D models in Fig. 6. The qualitative results indicate that, although Point-MF does not aim to generate textures or dense meshes, it can efficiently recover aligned object-level geometry from a single image. Additional qualitative comparisons on the remain-

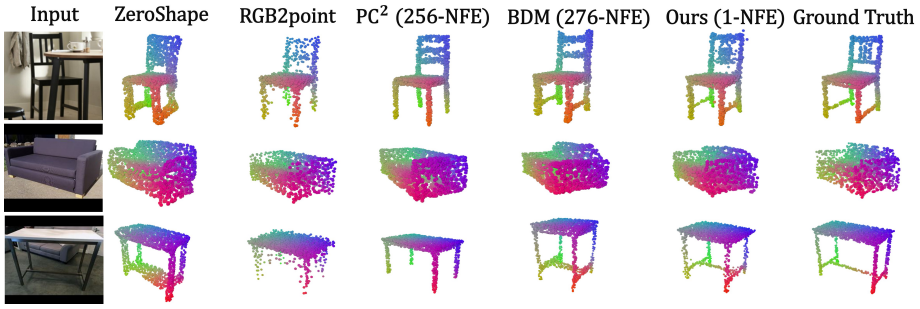


Fig. 5: Qualitative results on Pix3D.

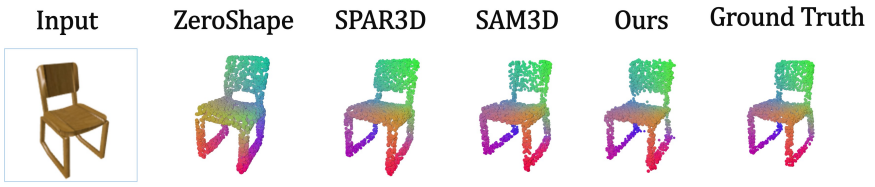


Fig. 6: Representative reference comparison with recent image-to-3D models on ShapeNet-R2N2. All outputs are visualized as point clouds from the same viewpoint.

ing ShapeNet-R2N2 categories and Pix3D, random samples, failure cases, and additional reference comparisons with recent image-to-3D models are provided in Supplementary Sec. E.

5.1 Ablations

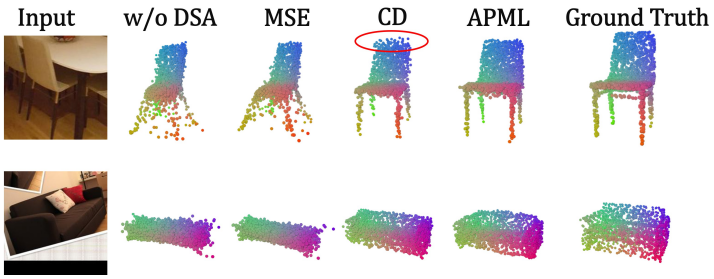


Fig. 7: Qualitative comparison of different set distances used in DSA on Pix3D. When Chamfer Distance is used, points tend to scatter slightly in some regions, as indicated by the red circles.

Tables 1 and 3 show that DSA is the most influential component, CTI provides consistent gains, and replacing DINOv3 with a standard ImageNet-pretrained ViT-B [10] causes only a small drop. Thus, the gain mainly comes from denoised-space anchoring and task-specific conditioning, not the image encoder alone. NFE–quality and diversity analyses support our 1-NFE setting and show retained output stochasticity under DSA (Supplementary Sec. D). We further ablate the set distance used in DSA on Pix3D by replacing APML with L1-Chamfer Distance (CD) or MSE. As shown in Table 5, APML achieves the best overall performance, whereas MSE performs the worst because DSA regularizes an unordered denoised point set without explicit point-to-point correspondences. Thus, set-to-set losses are more appropriate than point-wise MSE,

Table 5: Ablation study of the set distance used in DSA on Pix3D.

	CD×100 ↓				EMD×100 ↓			
Method	Chair	Sofa	Table	Mean	Chair	Sofa	Table	Mean
APML	6.48	**5.02**	**8.20**	**6.53**	**6.74**	**5.06**	**7.94**	**6.61**
CD	**6.36**	5.12	8.40	6.54	7.48	5.26	8.53	7.17
MSE	11.18	8.63	12.84	10.93	10.1	9.52	10.21	9.98
w/o DSA	8.82	6.58	10.12	8.57	8.88	7.02	9.76	8.62

and APML gives the best EMD-oriented trade-off. Fig. 7 further shows that CD can produce scattered points due to nearest-neighbor matching.

5.2 Limitations

Our method has three limitations. First, the self-consistent target computation in Mean Flow requires a JVP, which increases the computational cost and memory usage during training. Second, since our method operates directly in point-cloud space, the computational cost increases with the number of output points; obtaining higher-fidelity dense outputs can further benefit from external upsampling or refinement methods such as RepKPU [52]. Third, our evaluation is limited to ShapeNet-R2N2 and Pix3D, and validation on broader categories and more complex real-world images remains future work.

6 Conclusion and Discussion

In this paper, we formulated geometry-only point cloud reconstruction from a single image as one-step conditional Mean Flow directly in point-cloud space, and proposed Point-MF. Point-MF predicts an interval-averaged velocity field without using a point-cloud autoencoder and generates a point cloud with a single network function evaluation. We further introduced Denoised Space Anchor (DSA), which anchors the denoised estimate induced by the predicted mean velocity to the ground-truth shape using a set-distance loss, thereby suppressing outliers and density imbalance caused by large one-step updates.

Experiments on ShapeNet-R2N2 and Pix3D showed that Point-MF achieves strong CD, EMD, and F-Score against the evaluated point-cloud reconstruction baselines. In particular, Point-MF operates with only one NFE, while being substantially faster than iterative diffusion-based methods and maintaining the same order of inference latency as feedforward methods. These results demonstrate that geometric anchoring in the denoised space is effective for stabilizing one-step Mean Flow in point-cloud space.

Future work includes reducing the training cost associated with JVP, extending Point-MF to high-resolution point cloud generation, and validating it on broader categories and more complex real-world images. In addition, incorporating recent improvements from Improved Mean Flows [14] and Pixel Mean Flow [41] may further improve guidance and training objectives.

Acknowledgements This work was supported by JST CRONOS JPMJCS24K4 and JSPS KAKENHI 26K23875.

References

1. An, J., Yang, Z., Wang, J., Li, L., Liu, Z., Wang, L., Luo, J.: Bring metric functions into diffusion models. In: Proc. of the International Joint Conference on Artificial Intelligence. pp. 578–586 (2024)
2. Boss, M., Huang, Z., Vasishta, A., Jampani, V.: Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16240–16250 (2025)
3. Boulch, A., Marlet, R.: Poco: Point convolution for surface reconstruction. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6302–6314 (2022)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012 (2015)
5. Chen, H., Gu, J., Chen, A., Tian, W., Tu, Z., Liu, L., Su, H.: Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2416–2425 (2023)
6. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7220–7232 (2026)
7. Cho, J., Youwang, K., Yang, H., Oh, T.H.: Robust 3d shape reconstruction in zero-shot from a single image in the wild. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22786–22798 (2025)
8. Chou, G., Bahat, Y., Heide, F.: Diffusion-sdf: Conditional generative modeling of signed distance functions. In: Proc. of the IEEE/CVF International Conference on Computer Vision. pp. 2262–2272 (2023)
9. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: Proc. of the European Conference on Computer Vision. pp. 628–644 (2016)
10. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020)
11. Fang, H.S., Wang, C., Gou, M., Lu, C.: GraspNet-1Billion: A large-scale benchmark for general object grasping. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11444–11453 (2020)
12. Frans, K., Hafner, D., Levine, S., Abbeel, P.: One step diffusion via shortcut models. In: Proc. of the International Conference on Learning Representations (2025)
13. Geng, Z., Deng, M., Bai, X., Kolter, J.Z., He, K.: Mean flows for one-step generative modeling. In: Advances in Neural Information Processing Systems. vol. 38 (2025)
14. Geng, Z., Lu, Y., Wu, Z., Shechtman, E., Kolter, J.Z., He, K.: Improved mean flows: On the challenges of fastforward generative models. arXiv preprint arXiv:2512.02012 (2025)
15. Geng, Z., Pokle, A., Luo, W., Lin, J., Kolter, J.Z.: Consistency models made easy. In: Proc. of the International Conference on Learning Representations (2025)
16. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 216–224. IEEE (2018)

17. He, X., Zou, Z.X., Chen, C.H., Guo, Y.C., Liang, D., Yuan, C., Ouyang, W., Cao, Y.P., Li, Y.: Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14822–14833 (2025)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)
19. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3d. In: Proc. of the International Conference on Learning Representations (2024)
20. Huang, Z., Boss, M., Vasishta, A., Rehg, J.M., Jampani, V.: Spar3d: Stable point-aware reconstruction of 3d objects from single images. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16860–16870 (2025)
21. Huang, Z., Jampani, V., Thai, A., Li, Y., Stojanov, S., Rehg, J.M.: ShapeClipper: Scalable 3D shape learning from single-view images via geometric and clip-based consistency. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12912–12922 (2023)
22. Huang, Z., Johnson, J., Debnath, S., Rehg, J.M., Wu, C.Y.: Pointinfinity: Resolution-invariant point diffusion models. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10050–10060 (2024)
23. Huang, Z., Stojanov, S., Thai, A., Jampani, V., Rehg, J.M.: Zeroshape: Regression-based zero-shot shape reconstruction. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10061–10071 (2024)
24. Insafutdinov, E., Dosovitskiy, A.: Unsupervised learning of shape and pose with differentiable point clouds. In: Advances in Neural Information Processing Systems. vol. 31 (2018)
25. Jia, T., Yan, D., Hao, D., Li, Y., Zhang, K., He, X., Li, L., Wang, Y., Chen, J., Jiang, L., et al.: UltraShape 1.0: High-fidelity 3D shape generation via scalable geometric refinement. In: arXiv preprint arXiv:2512.21185 (2025)
26. Jignasu, A., Balu, A., Sarkar, S., Hegde, C., Ganapathysubramanian, B., Krishnamurthy, A.: SDFConnect: Neural implicit surface reconstruction of a sparse point cloud with topological constraints. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 5271–5279 (2024)
27. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv preprint arXiv:2305.02463 (2023)
28. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. In: Advances in Neural Information Processing Systems. vol. 35, pp. 26565–26577 (2022)
29. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 139–1 (2023)
30. Kim, D., Lai, C.H., Liao, W.H., Murata, N., Takida, Y., Uesaka, T., He, Y., Mitsufuji, Y., Ermon, S.: Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In: Proc. of the International Conference on Learning Representations (2024)
31. L Navaneet, K., Mandikal, P., Jampani, V., Babu, V.: Differ: Moving beyond 3d reconstruction with differentiable feature rendering. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
32. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19982–19992 (2026)

33. Lan, Y., Hong, F., Yang, S., Zhou, S., Meng, X., Dai, B., Pan, X., Loy, C.C.: ln3Diff: Scalable latent neural fields diffusion for speedy 3d generation. In: Proc. of the European Conference on Computer Vision. pp. 112–130. Springer (2024)
34. Lan, Y., Zhou, S., Lyu, Z., Hong, F., Yang, S., Dai, B., Pan, X., Loy, C.C.: GaussianAnything: Interactive point cloud latent diffusion for 3D generation. In: Proc. of the International Conference on Learning Representations (2025)
35. Lee, J.J., Benes, B.: Rgb2point: 3d point cloud generation from single rgb images. In: Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2952–2962 (2025)
36. Lee, K., Yu, S., Shin, J.: Decoupled meanflow: Turning flow models into flow maps for accelerated sampling. In: Proc. of the International Conference on Learning Representations (2026)
37. Lee, S., Lin, Z., Fanti, G.: Improving the training of rectified flows. Advances in Neural Information Processing Systems **37**, 63082–63109 (2024)
38. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: Proc. of the International Conference on Learning Representations (2023)
39. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: Proc. of the International Conference on Learning Representations (2023)
40. Lu, C., Song, Y.: Simplifying, stabilizing and scaling continuous-time consistency models. In: Proc. of the International Conference on Learning Representations (2025)
41. Lu, Y., Lu, S., Sun, Q., Zhao, H., Jiang, Z., Wang, X., Li, T., Geng, Z., He, K.: One-step latent-free image generation with pixel mean flows. In: arXiv preprint arXiv:2601.22158 (2026)
42. Luo, S., Hu, W.: Diffusion probabilistic models for 3d point cloud generation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2837–2845 (2021)
43. Melas-Kyriazi, L., Rupprecht, C., Vedaldi, A.: Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12923–12932 (2023)
44. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
45. Mo, S., Xie, E., Chu, R., Hong, L., Niessner, M., Li, Z.: Dit-3d: Exploring plain diffusion transformers for 3d shape generation. Advances in Neural Information Processing Systems **36**, 67960–67971 (2023)
46. Mu, Y., Zuo, X., Guo, C., Wang, Y., Lu, J., Wu, X., Xu, S., Dai, P., Yan, Y., Cheng, L.: Gsd: View-guided gaussian splatting diffusion for 3d reconstruction. In: Proc. of the European Conference on Computer Vision. pp. 55–72 (2024)
47. Murez, Z., Van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: Proc. of the European Conference on Computer Vision. pp. 414–431 (2020)
48. Navaneet, K., Mathew, A., Kashyap, S., Hung, W.C., Jampani, V., Babu, R.V.: From image collections to point clouds with self-supervised shape and pose networks. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1132–1140 (2020)
49. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv:2212.08751 (2022)

50. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)
51. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: Proc. of the IEEE/CVF International Conference on Computer Vision. pp. 4195–4205 (2023)
52. Rong, Y., Zhou, H., Xia, K., Mei, C., Wang, J., Lu, T.: RepKPU: Point cloud upsampling with kernel point representation and deformation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21050–21060 (2024)
53. Sharifipour, S., Casado, C.Á., Sabokrou, M., López, M.B.: APML: adaptive probabilistic matching loss for robust 3d point cloud reconstruction. In: Advances in Neural Information Processing Systems. vol. 38 (2025)
54. Shin, S., Kim, D., Lim, H.: Adaptive 3d reconstruction via diffusion priors and forward curvature-matching likelihood updates. Advances in Neural Information Processing Systems **38**, 128819–128856 (2025)
55. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., et al.: Dinov3. arXiv:2508.10104 (2025)
56. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: Proc. of the International Conference on Machine Learning. pp. 2256–2265 (2015)
57. Song, Y., Dhariwal, P.: Improved techniques for training consistency models. In: Proc. of the International Conference on Learning Representations (2024)
58. Song, Y., Dhariwal, P., Chen, M., Sutskever, I.: Consistency models. In: Proc. of the International Conference on Machine Learning (2023)
59. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in Neural Information Processing Systems **32** (2019)
60. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: Proc. of the International Conference on Learning Representations (2021)
61. Sun, X., Wu, J., Zhang, X., Zhang, Z., Zhang, C., Xue, T., Tenenbaum, J.B., Freeman, W.T.: Pix3D: Dataset and methods for single-image 3d shape modeling. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2974–2983 (2018)
62. Thai, A., Stojanov, S., Upadhya, V., Rehg, J.M.: 3d reconstruction of novel object shapes from single images. In: Proc. of the International Conference on 3D Vision. pp. 85–95 (2021)
63. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proc. of the European Conference on Computer Vision. pp. 52–67 (2018)
64. Wang, Z., Wang, Y., Chen, Y., Xiang, C., Chen, S., Yu, D., Li, C., Su, H., Zhu, J.: Crm: Single image to 3d textured mesh with convolutional reconstruction model. In: Proc. of the European Conference on Computer Vision. pp. 57–74 (2024)
65. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., et al.: Native and compact structured latents for 3d generation. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14419–14429 (2026)
66. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proc.

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21469–21480 (2025)
67. Xu, H., Lei, Y., Chen, Z., Zhang, X., Zhao, Y., Wang, Y., Tu, Z.: Bayesian diffusion models for 3d shape reconstruction. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10628–10638 (2024)
68. Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., Neumann, U.: Point-nerf: Point-based neural radiance fields. In: Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5438–5448 (2022)
69. Yagubbayli, F., Wang, Y., Tonioni, A., Tombari, F.: Legoformer: Transformers for block-by-block multi-view 3d reconstruction. arXiv:2106.12102 (2021)
70. Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: LION: latent point diffusion models for 3d shape generation. In: Advances in Neural Information Processing Systems. pp. 10021–10039 (2022)
71. Zhang, H., Siarohin, A., Menapace, W., Vasilkovsky, M., Tulyakov, S., Qu, Q., Skorokhodov, I.: Alphaflow: Understanding and improving meanflow models. In: Proc. of the International Conference on Learning Representations (2026)
72. Zhou, L., Ermon, S., Song, J.: Inductive moment matching. In: Proc. of the International Conference on Machine Learning (2025)