PointInfinity: Resolution-Invariant Point Diffusion Models

Zixuan Huang[1,2*] Justin Johnson[1*] Shoubhik Debnath[1] James M. Rehg[2] Chao-Yuan Wu[1*]
[1]FAIR at Meta, [2]University of Illinois at Urbana-Champaign

Figure 1. We present a resolution-invariant point cloud diffusion model that trains at ***low-resolution*** (down to 64 points), but generates ***high-resolution*** point clouds (up to 131k points). This test-time resolution scaling ***improves*** our generation quality. We visualize our high-resolution 131k point clouds by converting them to a continuous surface.

Abstract

We present PointInfinity, an efficient family of point cloud diffusion models. Our core idea is to use a transformer-based architecture with a fixed-size, resolution-invariant *latent representation. This enables efficient training with low-resolution point clouds, while allowing high-resolution point clouds to be generated during inference. More importantly, we show that scaling the test-time resolution beyond the training resolution* improves *the fidelity of generated point clouds and surfaces. We analyze this phenomenon and draw a link to classifier-free guidance commonly used in diffusion models, demonstrating that both allow trading off fidelity and variability during inference. Experiments on CO3D show that PointInfinity can efficiently generate high-resolution point clouds (up to 131k points, 31× more than Point-E) with state-of-the-art quality.*

1. Introduction

Recent years have witnessed remarkable success in diffusion-based 2D image generation [6, 38, 39], characterized by unprecedented visual quality and diversity in gen-

*Work done at Meta.

erated images. In contrast, diffusion-based 3D point cloud generation methods have lagged behind, lacking the realism and diversity of their 2D image counterparts. We argue that a central challenge is the substantial size of typical point clouds: common point cloud datasets [11, 50] typically contain point clouds at the resolution of 100K or more. This leads to prohibitive computational costs for generative modeling due to the quadratic complexity of transformers with respect to the number of input points. Consequently, state-of-the-art models are severely limited by computational constraints, often restricted to a low resolution of 2048 or 4096 points [32, 36, 46, 57, 59].

In this paper, we propose an efficient point cloud diffusion model that is efficient to train and easily scales to high resolution outputs. Our main idea is to design a class of architectures with fixed-sized, *resolution-invariant* latent representations. We show how to efficiently train these models with low resolution supervision, while enabling the generation of high-resolution point clouds during inference.

Our intuition comes from the observation that different point clouds of an object can be seen as different samples from a shared continuous 3D surface. As such, a generative model that is trained to model multiple low-resolution samples from a surface ought to learn a representation from the underlying surface, allowing it to generate high-resolution samples after training.

To encode this intuition into model design, we propose to decouple the representation of the underlying surface and the representation for point cloud generation. The former is a constant-sized memory for modeling the underlying surface. The latter is of variable size, depending on point cloud resolution. We design lightweight read and write modules for communicating between the two representations. The bulk of our model's computation is spent on modeling the underlying surface.

Our experiments demonstrate a high level of resolution invariance with our model[1]. Trained at a low resolution of 1,024, the model can generate up to 131k points during inference with state-of-the-art quality, as shown in Fig. 1. Interestingly, we observe that using a higher resolution than training in fact leads to slightly **higher** surface fidelity. We analyze this intriguing phenomenon and draw connection to classifier-free guidance. We emphasize that our generation output is $>30\times$ higher resolution than those from Point-E [36]. We hope that this is a meaningful step towards scalable generation of *high-quality* 3D outputs.

2. Related Work

Single-view 3D reconstruction aims to recover the 3D shape given an input image depicting an object or a scene. Recent works can be categorized based on the 3D representation they choose. Commonly used representation includes point clouds [8], voxels [5, 12, 54], meshes [13, 49] and implicit representations [33, 55]. Results of these works are usually demonstrated on synthetic datasets and/or small-scale real-world datasets such as Pix3D [45]. More recently, MCC [51] proposes to predict occupancy using a transformer-based model. It shows great zero-shot generalization performance, but it fails to model fine surface details due to its distance-based thresholding [51]. Our formulation avoids this issue and generates more accurate point clouds. Also note that most prior works are regression-based, which leads to deterministic reconstruction, ignoring the multi-modal nature of the reconstruction problem. Our diffusion-based method generates diverse outputs.

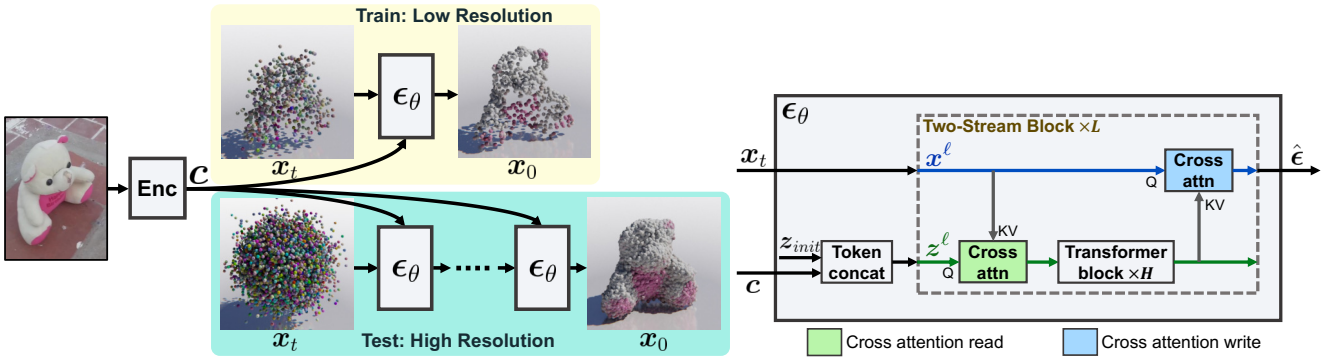
Generative 3D modeling learns the distribution of 3D assets, instead of a deterministic mapping. Early approaches in this direction often consider modeling 3D generation with GAN [1, 2, 9, 18, 27, 43, 47, 52], normalizing flow [24, 26, 56] or VAE [10, 34, 53]. More recently, with the success of 2D diffusion models [6, 38], diffusion-based 3D generative models [3, 4, 17, 28, 30, 35, 42, 44, 58] have been proposed and achieve promising generation quality. Among 3D diffusion models, point cloud diffusion models [32, 36, 46, 57, 59] are the most relevant ones to our work. We share the same diffusion framework with these approaches, but propose a novel resolution-invariant method that is both accurate and efficient. We also goes beyond noise-free synthetic datasets and demonstrate success on more challenging real-world datasets such as CO3D [37].

Transformers are widely used in various domains in computer vision [7, 29]. We extend transformers to use a fixed-sized latent representation for a resolution-invariant modeling of 3D point clouds. The resulting family of architectures includes architectures used in some prior works in recognition and 2D generation [19–21], that were originally designed for joint modeling of multiple modalities.

3. Background

Problem Definition. The problem studied in this work is RGB-D conditioned point cloud generation, similar to MCC [51]. Formally, we denote RGB-D images as $I \in \mathbb{R}^{4\times h\times w}$ and point clouds as $\boldsymbol{p} \in \mathbb{R}^{n\times 6}$, with 3 channels for RGB and 3 for XYZ coordinates. The point clouds we consider in this work can come from various data sources, including the noisy ones from multi-view reconstruction algorithms [37].

[1]The resolution-invariance discussed in this paper refers to the property we observe empirically as in experiments, instead of a strict mathematical invariance



(a) **PointInfinity Overview** (b) **Denoiser Architecture**

Figure 2. **Conditional 3D Point Cloud Generation with PointInfinity.** (a): At the core of PointInfinity is a resolution-invariant conditional denoising model ϵ_θ. It uses low-resolution point clouds for training and generates high-resolution point clouds at test time. (b): The main idea is a "Two-Stream" transformer design that decouples a fixed-sized latent representation $\boldsymbol{z}$ for capturing the underlying 3D shape and a variable-sized data representation $\boldsymbol{x}$ for modeling of the point could space. 'Read' and 'write' cross-attention modules are used to communicate between the two streams of processing. Note that most of the computation happens in the *latent stream* for modeling the underlying shape. This makes it less susceptible to the effects of point cloud resolution variations.

Denoising Diffusion Probabilistic Model (DDPM). Our method is based on the DDPM [15], which consists of two processes: 1) the diffusion process which destroys data pattern by adding noise, and 2) the denoising process where the model learns to denoise. At timestep $t \in [0, T]$, the diffusion process blends Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with data sample $\boldsymbol{p}_0$ as

$$\boldsymbol{p}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{p}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{1}$$

where $\bar{\alpha}_t$ denotes the noise schedule. The denoiser $\boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t)$ then learns to recover the noise from $\boldsymbol{p}_t$ with loss

$$L_{simple}(\theta) = \mathbb{E}_{t,\boldsymbol{p}_0,\boldsymbol{\epsilon}}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t)\|_2^2. \tag{2}$$

During inference, we use the stochastic sampler proposed in Karras et al. [23] to generate samples.

Classifier-Free Guidance. Conditional diffusion models often use classifier-free guidance [14] to boost the sample quality at the cost of sample diversity. During training, the condition of the model is dropped with some probability and the denoiser will learn to denoise both with and without condition. At test time, we linearly combine the conditional denoiser with unconditional denoiser as follows

$$\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{p}_t, t|\boldsymbol{c}) = (1+\omega)\boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t|\boldsymbol{c}) - \omega\boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t), \tag{3}$$

where ω is the classifier-free guidance scale and $\tilde{\boldsymbol{\epsilon}}_\theta(\boldsymbol{p}_t, t|\boldsymbol{c})$ is the new denoiser output.

Transformer-based [48] point diffusion models have been widely used in prior works [36], due to its permutation equivariant nature. Namely, when we permute the input noisy point cloud, transformers guarantee that the output noise predictions are also permuted in the same way.

However, as we will show in §5, vanilla transformers are not resolution-invariant — Testing with a different resolution from training significantly reduces accuracy. Furthermore, they scale quadratically w.r.t. to resolution, making them unamenable for high-resolution settings. To generate denser outputs, Point-E [36] trains a separate upsampler for upsampling points from 1024 to 4096. In the next section, we will show how to scale the resolution to up to 131k points without a separate upsampler.

4. Point Cloud Generation with PointInfinity

The main idea of PointInfinity is a resolution-invariant model, with which we train the model efficiently using low-resolution point clouds, while still supporting point cloud generation at a higher resolution. Fig. 2 illustrates an overview of the system.

4.1. Model

To achieve resolution invariance, we propose to parameterize $\epsilon_\theta(\boldsymbol{p}_t, t|c)$ to be a *2-stream* transformer-based model. The model first linearly projects noisy input points $\boldsymbol{p}_t$ into representations $\boldsymbol{x}_t$. Then a stack of L two-stream blocks process $\boldsymbol{x}_t$ and finally predicts $\hat{\boldsymbol{\epsilon}}$.

The Two-Stream Block. The main idea of our two-stream block is to introduce a fixed-sized latent representation $\boldsymbol{z}$ for capturing the underlying 3D shape and a *latent* processing stream for modeling it. Concretely, the ℓ-th block takes in two inputs $\boldsymbol{x}^\ell \in \mathbb{R}^{n\times d}$, $\boldsymbol{z}^\ell \in \mathbb{R}^{m\times d}$ and outputs $\boldsymbol{x}^{(\ell+1)} \in \mathbb{R}^{n\times d}$, $\boldsymbol{z}^{(\ell+1)} \in \mathbb{R}^{m\times d}$. At the first two-stream block ($\ell = 0$), the data-stream $\boldsymbol{x}^0$ is fed with the noisy point cloud $\boldsymbol{x}_t$. The latent input of the first block $\boldsymbol{z}^0$ is a

learned embedding $\boldsymbol{z}_{\text{init}}$ cancatenated with conditioning tokens c in the token dimension.

Within each two-stream block, we will first use a *read* cross attention block to cross attend information from data representation $\boldsymbol{x}^\ell$ into the latent representation $\boldsymbol{z}^\ell$,

$$\tilde{\boldsymbol{z}}^\ell := \text{CrossAttn}(\boldsymbol{z}^\ell, \boldsymbol{x}^\ell, \boldsymbol{x}^\ell), \tag{4}$$

where $\text{CrossAttn}(Q, K, V)$ denotes a cross attention block with query Q, key K, and value V. Then we use H layers of transformer blocks to model the latent representation

$$\boldsymbol{z}^{(\ell+1)} := \text{Transformer}(\tilde{\boldsymbol{z}}^\ell) \tag{5}$$

Finally, we will use a *write* cross attention block to write the latent representation back into the data stream through

$$\boldsymbol{x}^{(\ell+1)} := \text{CrossAttn}(\boldsymbol{x}^\ell, \boldsymbol{z}^{(\ell+1)}, \boldsymbol{z}^{(\ell+1)}) \tag{6}$$

Fig. 2b illustrates our design. Note that the *latent stream* processes tokens that are fixed-sized, while the *data stream* processes variable-sized tokens projected from noisy point cloud data. Since the bulk of the computation is spent on the fixed-sized latent stream, the processing is less affected by the resolution of the data stream. Also note that with this design, the computation only grows linearly with the size of $\boldsymbol{x}$, instead of growing quadratically.

4.2. Implementation Details

Architecture Details. We use $L = 6$ two-stream blocks in our denoiser, each includes $H = 4$ transformer blocks. For conditioning, we use the MCC encoder [51] to encode the RGB-D image into 197 tokens, and we use the time step embedding in [36] to encode time step t as a vector. Concatenating these two along the token dimension, we obtain the condition tokens c consisting of 198 vectors of dimension $d = 256$. $\boldsymbol{z}_{\text{init}}$ consists of 256 tokens, so the latent representation $\boldsymbol{z}^\ell$ has $m = 454$ tokens in total. The default training resolution n_{train} we use is 1024, while the test-time resolution n_{test} we consider in the experiments varies from 1024 to 131,072.

Training Details. We train our model with the Adam [25] optimizer. We use a learning rate of 1.25×10^{-4}, a batch size of 64 and momentum parameters of (0.9, 0.95). We use a weight decay of 0.01 and train our model for 150k iterations on CO3D. For diffusion parameters, we use a total of 1024 timesteps with the cosine noise scheduler. We also use latent self-conditioning of probability 0.9 during training following [19].

Surface Extraction. Because our model is able to generate high-resolution point clouds, it is possible to directly extract surface from the generated point clouds. To do so, we first create a set of 3D grid points in the space. For each point, we find the neighbor points in the point cloud and compute the mean distance to these points. We then use the marching cube [31] to extract the surface by thresholding the mean distance field.

5. Experiments

5.1. Dataset

CO3D. We use CO3D-v2 [37] as our main dataset for experiments. CO3D-v2 is a large real-world collection of 3D objects in the wild, that consists of ∼37k objects from 51 object categories. The point cloud of each object is produced by COLMAP [40, 41] from the original video capture. Despite the noisy nature of this process, we show that our model produces faithful 3D generation results.

5.2. Evaluation Protocol

Metrics. Following [16, 33, 51], the main evaluation metric we use for RGB-D conditioned shape generation is Chamfer Distance (CD). Given the predicted point cloud S_1 and the groundtruth point cloud S_2, CD is defined as an average of accuracy and completeness:

$$d(S_1, S_2) = \frac{1}{2|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2 + \frac{1}{2|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x - y\|_2 \tag{7}$$

Another metric we consider is F-score, which measures the alignment between the predicted point cloud and the groundtruth under a classification framing. Intuitively, it can be understood as the percentage of surface that is correctly reconstructed. In our work, we use a threshold of 0.2 for all experiments — if the distance between a predicted point and a groundtruth point is less than 0.2, we consider it as a correct match.

In addition to shape evaluation metrics, we also consider peak signal-to-noise ratio (PSNR) for texture evaluation.

Protocol. Note that point clouds with more points might be trivially advantageous in *completeness*, and thus Chamfer Distance or F-score. Consequently, in this paper we compute CD not only on the traditional *full point cloud* setting (denoted 'CD@full'), but also the *subsampled* setting (1024 points by default; denoted 'CD@1k') to ensure all methods are compared under the same number of points. Intuitively, 'CD@1k' measures the 'surface quality' under a certain resolution.[2] In addition, all objects are standardized such that they have zero mean and unit scale to ensure a balanced evaluation across all objects.

[2]For F-score, we always report the subsampled version.

5.3. Baselines

We compare PointInfinity with two SOTA models, Multi-view Compressive Coding (MCC) [51] and Point-E [36].

MCC [51] studies the problem of RGB-D conditioned shape reconstruction and learns implicit reconstruction with regression losses. MCC and our model use the same RGB-D encoder and both use CO3D-v2 as training set. One main difference between MCC and our model is that MCC uses a deterministic modeling and does not model interactions between query points.

Point-E [36] is a point cloud diffusion model using a vanilla transformer backbone. As the official training code is not released, we report results based on our reimplementation. We use the same RGB-D encoder as our method for fair comparison. The main difference between Point-E and PointInfinity lies the architecture of the diffusion denoisers.

5.4. Main Results

Test-Time Resolution Scaling. Table 1 compares performance of PointInfinity at different testing resolutions n_{test}. As we can see, despite that the $n_{\text{test}} \neq n_{\text{train}}$, increasing test-time resolution in fact slightly *improves* the generated surface quality, as reflected on CD@1k. This verifies the resolution invariance property of PointInfinity. We hypothesize the slight improvement comes from that the read operator gets to incorporate more information into the latent representation, leading to better modeling of the underlying surface. In §6, we will provide a more detailed analysis. On the contrary, the performance of Point-E [36] *decreases* with higher testing resolution. This is expected, as unlike PointInfinity, the size of Point-E [36]'s latent representations changes with the resolution, affecting the behavior of all attention operations, making it *not* resolution-invariant.

Generalization Analysis. Here we analyze how PointInfinity generalizes to different settings like different conditions and backbones. Table 2 presents results on a different condition. Specifically, we explore whether our finding generalizes to the "RGB-conditioned" point generation task. We can see that when only conditioned on RGB images, PointInfinity similarly demonstrates strong resolution invariance. Performance evaluated on all three metrics improves as test-time resolution n_{test} increases.

Note that our default implementation based on [19] represents only one instance of the two-stream family. The PerceiverIO [20] architecture originally designed for fusing different input modalities for recognition is another special case of a two-stream transformer model. The main difference between our default architecture and PerceiverIO lies in the number of read-write cross attention. Table 3 presents scaling behaviors with PerceiverIO. We can see that as expected, the performance similarly improves as the test-time resolution increases. This verifies that our findings generalize to other backbones within the two-stream family.

Metric	Method	1024	2048	4096	8192
CD@1k (↓)	Point-E [36]	0.239	0.213	0.215	0.232
	Ours	**0.227**	**0.197**	**0.186**	**0.181**
CD@full (↓)	Point-E [36]	0.239	0.200	0.194	0.205
	Ours	**0.227**	**0.185**	**0.164**	**0.151**
PSNR (↑)	Point-E [36]	13.31	13.46	13.28	12.60
	Ours	**13.37**	**13.88**	**14.15**	**14.27**

Table 1. **Effect of Test-Time Resolution Scaling.** Here we compare PointInfinity and Point-E [36] at different testing resolutions n_{test}. With PointInfinity, using a higher resolution during testing does not only lead to denser capture of the surface, it also improves the surface quality, as reflected by CD@1k and PSNR. On the contrary, Point-E, which uses a vanilla transformer backbone, sees a performance drop at high resolution.

Resolution	1024	2048	4096	8192
CD@1k (↓)	0.405	0.372	0.352	**0.343**
FS (↑)	0.336	0.376	0.398	**0.409**
PSNR (↑)	10.94	11.39	11.63	**11.75**

Table 2. **Generalization to the RGB condition.** Here we evaluate PointInfinity trained only with RGB condition at different testing resolutions n_{test}. We observe a similar performance improving trend with higher test-time resolutions.

Resolution	1024	2048	4096	8192
CD@1k (↓)	0.251	0.213	0.203	**0.197**
CD@full (↓)	0.251	0.199	0.177	**0.163**
PSNR (↑)	13.09	13.63	13.85	**13.97**

Table 3. **Generalization to Different Backbone Variants.** Our two-stream transformer design include a wide range of variants, including the PerceiverIO [20] architecture originally designed for fusing different input modalities for recognition. We observe a similar performance-improving property of test-time resolution scaling with this backbone variant as well.

SOTA Comparisons. We then compare PointInfinity with other state-of-the-art methods on CO3D, including MCC [51] and Point-E [36]. We report the result under a test-time resolution of 16k for our method. As shown in Table 4, our model outperforms other SOTA methods significantly. PointInfinity achieves not only better surface generation fidelity (9% better than Point-E and 24% better than MCC quantified by CD@1k), but also generates better texture (as shown in better PSNR).

Method	CD@1k (↓)	FS (↑)	PSNR (↑)
MCC [51]	0.234	0.549	14.03
Point-E [36]	0.197	0.675	14.25
PointInfinity	**0.179**	**0.724**	**14.31**

Table 4. **Comparison with Prior Works.** We see that PointInfinity outperforms other state-of-the-art methods significantly on all metrics we evalute, demonstrating the effectiveness our resolution-invariant point diffusion design.

Comparisons with Unconditional Models. Additionally, we compare PointInfinity with unconditional 3D generative models in terms of resolution-invariance. Specifically, we consider Point-Voxel Diffusion (PVD) [32] and Gradient Field (ShapeGF) [2]. These models are originally designed for unconditional 3D shape generation (no color), and are trained with different resolutions and data. Therefore, we report relative metrics when comparing with them, so that numbers between different methods are comparable. The results of relative CD are shown in Tab. 5. We observe that as resolution increases, PointInfinity's performance improves, while ShapeGF's performance remains almost unchanged. On the other hand, PVD's performance significantly drops. This verifies the superior resolution-invariance property of PointInfinity, even when compared to models designed for different 3D generation scenarios.

Resolution	1×	2×	4×	8×
PVD [32]	1.000	3.605	4.290	4.221
GF [2]	1.000	0.999	1.000	0.999
PointInfinity	1.000	**0.868**	**0.819**	**0.797**

Table 5. **Comparison with Unconditional Models.** We see that PointInfinity outperforms other unconditional 3D generative methods, including PVD and ShapeGF, in terms of resolution-invariance.

5.5. Complexity Analysis

We next analyze the computational complexity of PointInfinity at different test-time resolutions. The computational analysis in this section is performed on a single NVIDIA GeForce RTX 4090 GPU with a batch size of 1. Thanks to the resolution-invariance property, PointInfinity can generate point clouds of different test-time resolutions n_{test} without training multiple models. On the other hand, Point-E [36] requires the training resolution to match with the testing resolution, since it is resolution specific. We present detailed benchmark results comparing the iteration time and memory for both training and testing in Fig. 3. We can see that the training time and memory of Point-E model scales *quadratically* with test-time resolution, while our model remains *constant*. Similarly at test time, Point-E scales quadratically with input resolution, while our inference computation scales *linearly*, thanks to our two-stream design.

We further compare the computational efficiency of PointInfinity to diffusion models with implicit representations. We consider the state-of-the-art implicit model, Shap-E [22]. For a comprehensive comparison, we run Shap-E under different commonly used marching cubes resolutions and show results in Fig. 4. Our results show that PointInfinity is faster and more memory-efficient than Shap-E.

Overall, PointInfinity demonstrates significant advantage in computational efficiency.

5.6. Ablation Study

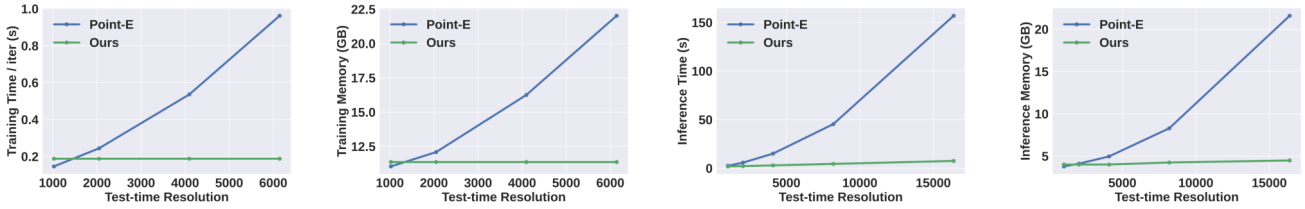
Training Resolution. In Table 6a, we train our model using different training resolutions and report the performance under a test-time resolution of 16k. We can see that PointInfinity is insensitive to training resolutions. We choose 1024 as our training resolution to align with Point-E [36].

Number of Latent Tokens. We next study the impact of representation size (the number of tokens) used in the 'latent stream'. As shown in Table 6b, 256 or higher tends to provide strong results, while smaller values are insufficient to model the underlying shapes accurately. We choose 256 as our default latent token number for a good balance between performance and computational efficiency.

Comparison to A Naïve Mixture Baseline. Finally, note that a naïve way to increase testing resolution without retraining a model is to perform inference multiple times and combine the results. We compare PointInfinity with the naïve mixture baseline (denoted 'mixture') in Table 6c. Interestingly, we observe that the mixture baseline sees a slight improvement with higher resolutions, instead of staying constant. In a more detailed analysis we found that mixing multiple inference results reduces the bias and improves the overall coverage, and thus its CD@1k and FS. Nonetheless, PointInfinity performs significantly better, verifying the non-trivial modeling power gained with our design. Also note that PointInfinity is significantly more efficient, because all points share the same fixed-sized latent representation and are generated in one single inference run.

5.7. Qualitative Evaluation

Here we qualitatively compare PointInfinity with other state-of-the-art methods in Fig. 5. Compared to MCC [51], we observe that our method generates more accurate shapes and details, confirming the advantage of using a diffusion-based point cloud formulation. Compared to Point-E [36], PointInfinity is able to generate much denser (up to 131k) points, while Point-E generates up to 4k points, which are insufficient to offer a complete shape. When comparing under the same resolution, we observe that PointInfinity



(a) **Train Iter Time** (b) **Train Memory** (c) **Inference Iter Time** (d) **Inference Memory**

Figure 3. **PointInfinity scales favorably compared to Point-E [36] in both computation time and memory for both training and inference.** (a,b): Thanks to the resolution-invariant property of PointInfinity, the training iteration time and memory stays constant regardless of the test-time resolution n_{test}. Point-E on the other hand requires $n_{\text{train}} = n_{\text{test}}$ and scales quadratically. (c,d): Our inference time and memory scales linearly with respect to n_{test} with our two-stream transformer design, while Point-E scales quadratically with the vanilla transformer design.

n_{train}	CD@1k(↓)	FS(↑)	PSNR(↑)
64	0.178	0.722	14.28
256	0.174	0.737	14.41
1024 (default)	0.179	0.724	14.31
2048	0.183	0.708	14.19

(a) **Training Resolution**

z_{init} dim	CD@1k(↓)	FS(↑)	PSNR(↑)
64	0.457	0.262	10.90
128	0.182	0.719	14.25
256 (default)	0.179	0.724	14.31
512	0.176	0.729	14.45

(b) **Number of Latent Tokens**

	n_{test}	CD@1k(↓)	FS(↑)	PSNR(↑)
Mixture	1024	0.227	0.622	13.37
Mixture	2048	0.220	0.619	13.21
Mixture	4096	0.215	0.625	13.12
Mixture	8192	0.211	0.632	13.07
PointInfinity	8192	**0.181**	**0.721**	**14.27**

(c) **Mixture Baseline**

Table 6. **Ablation Experiments on CO3D-v2.** We perform ablations on the CO3D-v2 dataset [37]. Specifically, we study the impact of training resolution (a), the size of the latent representations (b), and verify the advantage of PointInfinity over a 'mixture' baseline for generating high resolution point clouds.

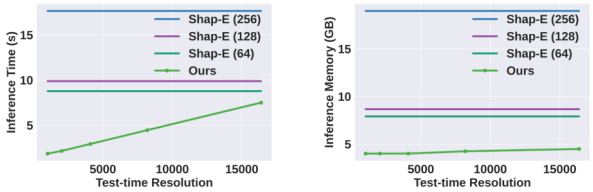


Figure 4. **PointInfinity achieves favorable computational complexity even compared with implicit methods such as Shap-E [22].** The figures show PointInfinity is faster and more memory-efficient than Shap-E under a high test-time resolution of 16k.

enjoys finer details and more accurate shapes than Point-E. Furthermore, We observe that PointInfinity not only achieves high-quality generation results in general, but the generated surface improves as the resolution increases.

6. Analysis

6.1. Mechanism of Test-time Resolution Scaling

In §5.4, we observe that test-time resolution scaling with PointInfinity improves the reconstruction quality. In this section, we provide a set of analysis to provide further insights into this property.

Recall that during diffusion inference, the model input is a linear combination of the Gaussian noise and the output from the previous sampling step. Our hypothesis is that, increasing the resolution results in a more consistent generation process, because more information are carried out between denoising steps. With a higher number of input tokens, the denoiser obtains strictly more information on previously denoised results $\boldsymbol{x}_t$, and thus $\boldsymbol{x}_{t-1}$ will follow the pattern in $\boldsymbol{x}_t$ better.

To verify this hypothesis, we consider a variant of our model, where the read module only reads from a fixed set of n_{train} input tokens. All other $n_{\text{test}} - n_{\text{train}}$ tokens' attention weights are set as zero. The remaining parts of the model are kept unchanged. As shown in Table 7, after this modification, CD@1k of the model does not improve with resolution anymore. Rather, it remains almost constant. This

Metric	Method	1024	2048	4096	8192
CD@1k (↓)	Restricted Read	0.227	0.225	0.220	0.224
	Default	0.227	**0.197**	**0.186**	**0.181**
CD@full (↓)	Restricted Read	0.227	0.211	0.196	0.190
	Default	0.227	**0.185**	**0.164**	**0.151**
PSNR (↑)	Restricted Read	13.37	13.39	13.50	13.49
	Default	13.37	**13.88**	**14.15**	**14.27**

Table 7. **Analysis of the Resolution Scaling Mechanism.** To verify our hypothesis discussed in §6, we compare our default implementation to a "Restricted Read" baseline, where the information intake is limited to 1024 tokens, at different test-time resolutions. We see that the performance no longer monotonically improves with resolution, supporting our hypothesis.

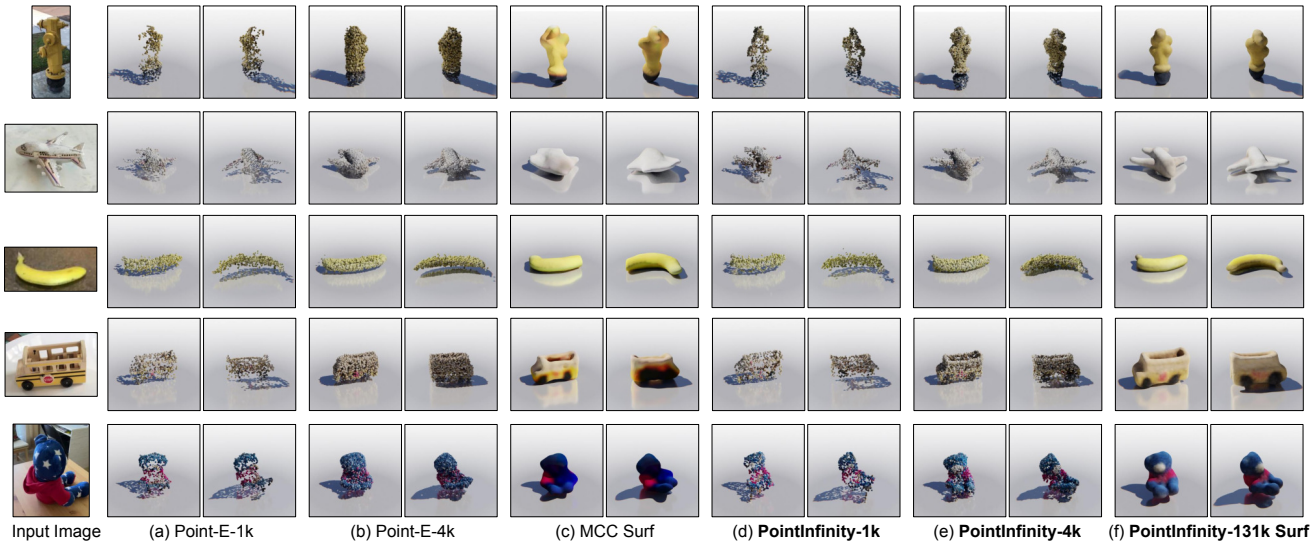


Figure 5. **Qualitative Evaluation on the CO3D-v2 Dataset [37].** The point clouds generated by our model (column d,e,f) represent denser and more faithful surfaces as resolution increases. On the contrary, Point-E (column a, b) does not capture fine details. In addition, we see that PointInfinity obtains more accurate reconstructions from the 131k-resolution point clouds (column f) compared to MCC's surface reconstructions (column c).

result supports that the high information intake indeed leads to performance improvement.

6.2. Variability Analysis

Based on our hypothesis, a potential side effect is a reduced variability, due to the stronger condition among the denoising steps. To verify this, we evaluate the variability of our sampled point clouds. Specifically, for every example in the evaluation set, we randomly generate 3 different point clouds and calculate the average of the pair-wise CD among them, as a measure of the variability. In Fig. 6, we see that when the resolution increases, the variability indeed reduces, supporting our hypothesis.

6.3. Comparison to Classifier-Free Guidance

The fidelity-variability trade-off observed in resolution scaling is reminiscent of the fidelity-variability trade-off often observed with classifier-free guidance [14]. We compare these two in Fig. 6. As we can see, when the guidance scale is small, classifier-free guidance indeed improves the fidelity at the cost of variability. However, when the guidance scale gets large, further increasing the guidance hurts the fidelity. On the contrary, our resolution scaling consistently improves the sample fidelity, even at very high resolution. Moreover, the trade-off achieved by PointInfinity is always superior to the trade-off of classifier-free guidance.

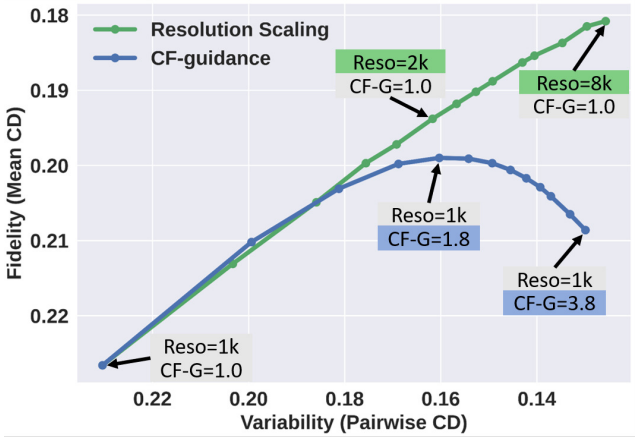


Figure 6. **Fidelity and Variability Analysis.** We observe that as the resolution increases, the variability of the generated point clouds reduces, due to the stronger condition among the denoising steps. Also note that our test-time resolution scaling achieves a better fidelity-variability trade-off than classifier-free guidance.

7. Conclusions

We present PointInfinity, a resolution-invariant point diffusion model that efficiently generates high-resolution point clouds (up to 131k points) with state-of-the-art quality. This is achieved by a two-stream design, where we decouple the latent representation for modeling the underlying shape and the point cloud representation that is variable in size. Interestingly, we observe that the surface quality in fact *improves* as the resolution increases. We thoroughly analyze this phenomenon and provide insights into the underlying mechanism. We hope our method and results are useful for future research towards scalable 3D point cloud generation.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 2

[2] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 364–381. Springer, 2020. 2, 6

[3] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023. 2

[4] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023. 2

[5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[6] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 1, 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[8] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[9] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 2

[10] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 40 (6):1–15, 2021. 2

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 2

[12] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14*, pages 484–499. Springer, 2016. 2

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3, 8

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[16] Zixuan Huang, Varun Jampani, Anh Thai, Yuanzhen Li, Stefan Stojanov, and James M Rehg. Shapeclipper: Scalable 3d shape learning from single-view images via geometric and clip-based consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12922, 2023. 4

[17] Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022. 2

[18] Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 397–413. Springer, 2020. 2

[19] Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022. 2, 4, 5

[20] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2021. 5

[21] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 2

[22] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 6, 7

[23] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022. 3

[24] Hyeongju Kim, Hyeonseung Lee, Woo Hyun Kang, Joun Yeop Lee, and Nam Soo Kim. Softflow: Probabilistic framework for normalizing flow on manifolds. *Advances in Neural Information Processing Systems*, 33:16388–16397, 2020. 2

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 4

[26] Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020. 2

[27] Chun-Liang Li, Manzil Zaheer, Yang Zhang, Barnabas Poczos, and Ruslan Salakhutdinov. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018. 2

[28] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023. 2

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[30] Zhen Liu, Yao Feng, Michael J Black, Derek Nowrouzezahrai, Liam Paull, and Weiyang Liu. Meshdiffusion: Score-based generative 3d mesh modeling. *arXiv preprint arXiv:2303.08133*, 2023. 2

[31] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. 1998. 4

[32] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2, 6

[33] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2, 4

[34] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 2

[35] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022. 2

[36] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2, 3, 4, 5, 6, 7

[37] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 2, 4, 7, 8

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[39] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 1

[40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 4

[41] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016. 4

[42] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20887–20897, 2023. 2

[43] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3859–3868, 2019. 2

[44] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2

[45] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[46] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. Gecco: Geometrically-conditioned point diffusion models. *arXiv preprint arXiv:2303.05916*, 2023. 2

[47] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *International conference on learning representations*, 2018. 2

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[49] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[50] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 2

[51] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compres-

sive coding for 3D reconstruction. *arXiv:2301.08247*, 2023. 2, 4, 5, 6

[52] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems*, 29, 2016. 2

[53] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2

[54] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2690–2698, 2019. 2

[55] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019. 2

[56] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 2

[57] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. *arXiv preprint arXiv:2210.06978*, 2022. 2

[58] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. *arXiv preprint arXiv:2305.04461*, 2023. 2

[59] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2