3DDM: Physically-based Anisotropic 3D Diffusion Model with 3D Gaussian for Point Cloud Completion

Long Xi[1,2*], Jia Ma[2], ZhenYu Yuan[2], Tao Xue[1], Wen Tang[3*], Wen Lv[1]
[1]School of Cybersecurity, Xi'an Polytechnic University, Xi'an, China
[2]School of Computer Science, Xi'an Polytechnic University, Xi'an, China
[3]Faculty of Media, Science and Technology, Bournemouth University, United Kingdom
{lxi, xuetao, 20240219}@xpu.edu.cn, {240721046, 240721124}@stu.xpu.edu.cn
wtang@bournemouth.ac.uk

Abstract

A 3D point cloud completion task is to generate completed 3D objects given partial observations. Auto-encoder-based models suffer from poor generalization ability to untrained 3D data. Current diffusion-based models add isotropic noise with the same variance in three x, y, z axes. More importantly, these models ignore real-world anisotropic evolution properties of 3D particles from a non-equilibrium state to thermodynamic equilibrium in the real physical world due to the velocity and energy thermodynamics of the particles, leading to unstable completions of 3D object topology. This paper presents a novel physically-based anisotropic 3D diffusion model (3DDM) to address these issues. We also present derivations of our proposed forward and reverse processes and a loss function in closed form, thus reproducibility. The 3DDM contains anisotropic energy-aware forward and reverse processes with a novel anisotropic quadratic loss function. The forward process adds anisotropic 3D Gaussian noises per-axis and mimics the thermal non-equilibrium evolution towards Maxwellian equilibrium based on velocity and kinetic energy evolutions of 3D particles in the real physical space. The reverse process learns to denoise along per-axis and per-timestep anisotropically. The anisotropic quadratic loss function penalizes errors along certain axes, yielding a highly flexible and anisotropic reverse diffusion process and a physically realistic generative model. The 3DDM denoises along x, y, z axes with different velocities from the non-equilibrium evolution, achieving fewer than 20 diffusion steps and strong generalization to unseen 3D objects and real-world scenes that were not trained.

Code and technical appendix —
https://github.com/LONG-XI/3DDM

Introduction

3D point clouds have been widely applied in various computer vision applications, including autonomous driving (Chen et al. 2025; Gu et al. 2025), game (Ji et al. 2025), robotic perception (Pan et al. 2025), and virtual reality (Novotny and Laidlaw 2024). 3D point cloud completions from partial observations are an active research topic.

Although most state-of-the-art auto-encoder-based networks, such as PCN (Yuan et al. 2018), FoldingNet (Yang et al. 2018), TopNet (Tchapmi et al. 2019), PoinTr (Yu et al. 2021), AnchorFormer (Chen et al. 2023), SeedFormer (Zhou et al. 2022) and FSC (Wu et al. 2024), demonstrate strong reconstruction performance on partial inputs, they perform poorly on generalization ability for untrained datasets. Quite often, these untrained datasets contain point distributions and densities that are different from the training datasets.

Current diffusion-based methods, such as PVD (Zhou, Du, and Wu 2021), SDFusion (Cheng et al. 2023), directly adopt the standard diffusion model in DDPM (Denoising Diffusion Probabilistic Model) (Ho, Jain, and Abbeel 2020). However, the DDPM adds isotropic noise with the same variance in x, y, z axes and does not realistically simulate the evolution of 3D particles from a non-equilibrium state to an equilibrium state in a real 3D physical space, ignoring the real-world anisotropic properties and real physical evolution of the 3D particles which is due to the velocity and the energy evolution of the particles. As a result, current DDPM-based models generate unstable 3D object topologies and require hundreds or even thousands of diffusion steps to converge, leading to low sampling efficiency and poor generalization ability for untrained datasets.

In this paper, we propose a novel physically-based anisotropic 3D diffusion model, 3DDM, to improve the generalization ability of the diffusion model and largely reduce the diffusion steps. We make an important observation: *the physically realistic modelling of the evolution of 3D particles from a non-equilibrium state to a thermodynamic equilibrium state in a real-world 3D physical space can enable the diffusion process to be accurately explained through a physically-based simulation process using closed forms. The simulation process can detect real-world anisotropies and improve the generalization ability for untrained datasets. Furthermore, the proposed physically-based 3DDM will require fewer diffusion steps for convergence since it learns to denoise anisotropically along x, y, z axes with different velocities from the non-equilibrium evolution.*

3DDM contains anisotropic energy-aware forward and reverse processes with a novel anisotropic quadratic loss. Specifically, we design three modules in the forward process of the 3DDM, including a kinetic energy evolution, an energy-aware β_t rescaling, and an anisotropic 3D Gaussian modeling. The kinetic energy evolution module mimics the thermal non-equilibrium evolution towards Maxwellian

*Corresponding author

equilibrium based on velocity and kinetic energy evolutions of 3D particles by adopting Langevin dynamics (Giovanni and Michele 2007). The energy-aware β_t rescaling module adaptively adjusts the diffusion strength based on per-axis energy deviations and guides axis-specific noise adjustment to Maxwell equilibrium. The anisotropic 3D Gaussian modeling module adds anisotropic 3D Gaussian noise per-axis, and this anisotropic diffusion makes the diffusion process physically-based and realistic.

The reverse process of 3DDM mimics the inverse processes of anisotropic thermalization in real physical systems, and learns to denoise along per-axis and per-timestep anisotropically. To further enhance training stability and physical consistency, we derive an anisotropic quadratic loss, which penalizes errors along certain axes, yielding a highly directional flexibility of the reverse diffusion process.

The main contributions of this paper are as follows:

- A novel physically-based anisotropic 3D diffusion model, 3DDM, for 3D point cloud completion with fewer diffusion steps.
- An anisotropic energy-aware forward process in 3DDM, which adds anisotropic 3D Gaussian noises per-axis and mimics the thermal non-equilibrium evolution towards Maxwellian equilibrium, enhancing the interpretability of the diffusion model.
- An anisotropic reverse process, which learns the reverse thermal non-equilibrium evolution and enables the model to capture anisotropic features, enhancing completion quality and generalization ability.
- Closed forms for the proposed forward, the reverse and the loss function.
- State-of-the-art completion performance and strong generalization ability to unseen real-world 3D semantic scenes that were never trained.

Related Work

Auto-encoder-based methods often adopt the PointNet-based (Qi et al. 2017a) encoder and proposed different decoders for reconstructing the global features of partial observations to complete 3D point clouds, such as a folding operation to deform 2D grids into 3D point clouds in the decoder (Yang et al. 2018), and a two-stage PointNet-based encoder (Yuan et al. 2018) for feature learning and uses fully connected layers followed by the folding operation to achieve completion results. TopNet (Tchapmi et al. 2019) designs a tree-based decoder where all leaf nodes generate a complete 3D point cloud. These methods focus on learning global features without considering local features for fine details.

Transformer-based methods consider the fine detailed local features for high-quality completion results. PoinTr (Yu et al. 2021) considers the completion process as a point-to-point translation problem and uses a transformer encoder-decoder to learn local features for fine details. SeedFormer (Zhou et al. 2022) and AnchorFormer (Chen et al. 2023) learns regional features for local pattern reconstruction. FSC (Wu et al. 2024) uses a dual-branch feature extractor with several attention modules for sparse point clouds. Although these methods perform well in reconstructing partial input, they are less effective in generating missing regions and lack structural completeness. Additionally, they commonly suffer from overfitting to the training distribution and degraded performance across different datasets, which shows poor generalization capabilities on unseen datasets that were never trained.

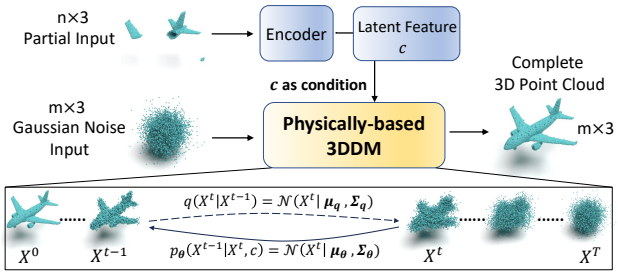


Figure 1: 3D point cloud completion architecture.

Originally developed for 2D image generation, diffusion models, from DDPM (Ho, Jain, and Abbeel 2020) in data space to LDM (Rombach et al. 2022) in latent feature space, offer greater quality than GANs (Goodfellow et al. 2014). Diffusion-based 3D point cloud completion method, PVD (Zhou, Du, and Wu 2021), uses the standard DDPM (Ho, Jain, and Abbeel 2020) directly in data space for 3D point cloud completion. SDFusion (Cheng et al. 2023) uses the standard LDM (Ho, Jain, and Abbeel 2020) in latent space for 3D signed distance function completion. However, these diffusion-based 3D methods rely on isotropic noise assumptions and do not realistically simulate the evolution of 3D particles from a non-equilibrium state to equilibrium in 3D physical space, which ignores real-world physical anisotropies in 3D point clouds and limits their physical realism and interoperability. As a result, they treat three spatial axes x, y, z uniformly and fail to capture the directional variations in real-world physical space, leading to unstable 3D structure completion with hundreds or thousands of diffusion steps. Additionally, the lack of a physical realism diffusion process in these methods severely limits their generalization ability to untrained datasets.

In contrast, our physically-based 3DDM model incorporates an anisotropic diffusion process based on velocity and kinetic energy evolutions, which realistically simulates the evolution of the non-equilibrium towards Maxwellian equilibrium. Our novel approach significantly improves completion accuracy on unseen geometric structures and cross-domain generalization, while achieving high-quality completion results with fewer than 20 diffusion steps.

Methods

3D point clouds often have geometries that vary along different axes. (e.g., A 3D table is wider in x/y and short in z. A 3D lamp is tall in z and narrow in x/y). The directional diffusion control reflects these anisotropies, which makes the

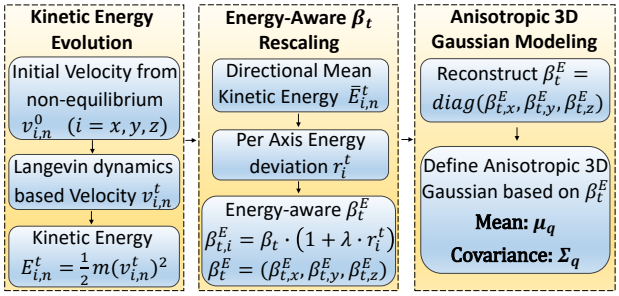


Figure 2: 3DDM forward process.

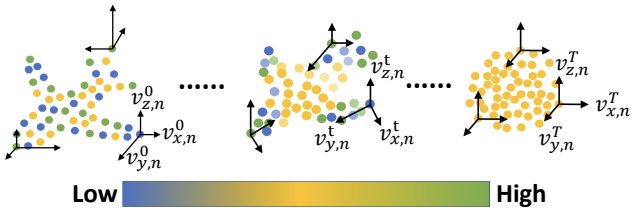
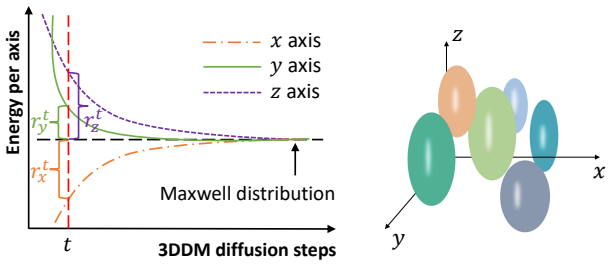


Figure 3: Kinetic energy and velocity evolutions in the forward process. Blue and green indicate the low and the high energies at the non-equilibrium state. Yellow indicates the thermal equilibrium.



(a) Per-axis energy deviations r_i^t (b) Contour surfaces of 3D Gaussian PDF in 3DDM

Figure 4: (a) Energy deviations r_i^t in energy-aware β_t rescaling. (b) 3D Gaussians in anisotropic 3D gaussian modeling.

diffusion process more physically realistic. When simulating realistic Langevin dynamics-based diffusion, the energy distribution often differs by axis.

The Maxwell-Boltzmann velocity distribution is a 3D Gaussian distribution, which illustrates the velocities of 3D particles in a Maxwellian equilibrium state in 3D physical space. The velocity distributions $f(v_x, v_y, v_z)$ of 3D particles are completely independent and uncorrelated along the x, y and z directions, as defined by Equation 1.

$$f(v_i) = \sqrt{\frac{m}{2\pi k_B T}} e^{-\frac{m v_i^2}{2 k_B T}}, \qquad i \in \{x, y, z\} \tag{1}$$

where m is the mass of a 3D particle; T is the absolute temperature; k_B is the Boltzmann constant; $V = (v_x, v_y, v_z)$ represents the velocities of 3D particles along the x, y and z directions, respectively.

Inspired by the Maxwell-Boltzmann velocity distribution in a physical equilibrium system, we propose a physically-based anisotropic 3D diffusion model, 3DDM, to mimic the forward and reverse processes of the thermal non-equilibrium evolution toward Maxwellian equilibrium physically and generate completion results from partial 3D point clouds. Figure 1 shows the overall architecture of 3DDM for 3D point cloud completion. Given a partial 3D point cloud input, where each point is defined as $P_i = (x, y, z)$. 3DDM considers latent features of the partial input as the condition and generates the corresponding complete 3D point cloud based on the condition. The 3DDM contains an anisotropic energy-aware forward process (Figure 2), which mimics the thermal non-equilibrium evolution toward Maxwellian equilibrium. The reverse process learns to denoise along per-axis and per-timestep anisotropically. The novel anisotropic quadratic loss function is proposed to train 3DDM.

Anisotropic Energy-Aware Forward Process

The novel anisotropic energy-aware forward process mimics the thermal non-equilibrium evolution towards Maxwellian equilibrium in 3D physical space. Three modules are proposed in the forward process: (i) a kinetic energy evolution; (ii) an energy-aware β_t rescaling; and (iii) anisotropic 3D Gaussian modeling modules, as shown in Figure 2.

Kinetic Energy Evolution. The kinetic energy evolution module aims to modulate the per-axis diffusion intensity by calculating the relative kinetic energy deviations of 3D particles from a thermal equilibrium state. First, we simulate the velocity evolution based on Langevin dynamics (Giovanni and Michele 2007), which models the relaxation of a non-equilibrium state towards a thermal equilibrium state, as defined by Equation 2.

$$\begin{aligned} v_i^0 &= v_i^{maxwell} + \sigma \cdot \xi_0, \; i \in \{x, y, z\}; \; \xi_0 \sim N(0, 1) \\ v_i^t &= (1 - \gamma) v_i^{t-1} + z\sqrt{2\gamma k_B T / m} \end{aligned} \tag{2}$$

where v_i^0 is the initial velocity of 3D particles for non-equilibrium perturbation. v_i^0 contains velocities (v_x^0, v_y^0, v_z^0) for the x, y and z directions, respectively. $v_i^{maxwell}$ contains velocities of N 3D particles sampled from a Maxwell-Boltzmann velocity distribution, which represents the thermal equilibrium. To mimic non-equilibrium conditions in real physical systems, a random Gaussian perturbation ξ_0 is added to the velocity of each particle. σ controls the magnitude of the disturbance. v_i^t and v_i^{t-1} denote the velocities in the steps t and $t-1$ along the i axis, respectively. γ is the damping coefficient. $(1-\gamma)v_i^{t-1}$ mimics physical damping, moving energy gradually from the non-equilibrium system towards the equilibrium. $z\sqrt{2\gamma k_B T/m}$ denotes stochastic kicks that represent thermal collisions with surrounding particles. The balance between these two terms ensures that the velocity distribution will stabilize towards a Maxwellian equilibrium over time.

We then define the kinetic energy evolution by Equation 3. As shown in Figure 3, the velocities and energies of each 3D point mimic the relaxation of a non-equilibrium state toward

thermal equilibrium.

$$E_{i,n}^t = \frac{1}{2} m (v_{i,n}^t)^2, \quad i \in \{x, y, z\} \tag{3}$$

where $E_{i,n}^t$ is the kinetic energy of a 3D particle along per axis at the timestep t in the forward process.

Energy-Aware β_t Rescaling. In the non-equilibrium state towards equilibrium, kinetic energy $E_{i,n}^t$ is anisotropically distributed. For example, 3D particles may move faster along the x-axis than along y and z axes. The energy-aware β_t rescaling module detects these anisotropic properties per axis and adjusts the noise along each axis based on kinetic energy deviations r_i^t in order to rebalance the system from an non-equilibrium state to a Maxwell equilibrium state, making the diffusion process more physically sound, realistic and directionally responsive. As shown in Figure 4 (a), r_i^t assesses how far the system is from the thermal equilibrium on a per-axis basis, and guides axis-specific noise adjustment towards a Maxwell equilibrium.

$$r_i^t = \frac{\bar{E}_{i,n}^t - E_k}{E_k}, \quad \bar{E}_{i,n}^t = \frac{1}{N} \sum_{n=1}^{N} E_{i,n}^t, \quad E_k = \frac{d}{2} k_B T \tag{4}$$

where $i \in \{x, y, z\}$. $\bar{E}_{i,n}^t$ is the mean kinetic energy computed along the x, y and z axes for N 3D points at each timestep, respectively. E_k is the equilibrium kinetic energy per axis.

r_i^t adaptively modifies the $\beta_{t,i}^E$ along each axis, where x, y, z axes can equilibrate at different rates. If the energy in the i axis is higher than the equilibrium, $\beta_{t,i}^E$ decreases, vice versa. The original scalar β_t in DDPM is rescaled to a per-axis version $\beta_{t,i}^E$, as defined by Equation 5.

$$\beta_{t,i}^E = \beta_t \cdot (1 + \lambda \cdot r_i^t), \quad \beta_t^E = (\beta_{t,x}^E, \beta_{t,y}^E, \beta_{t,z}^E) \tag{5}$$

where $i \in \{x, y, z\}$. λ controls how sensitive the system is to kinetic energy deviations. β_t^E is independent for x, y and z axes.

The energy-aware β_t rescaling module increases the diffusion density along axes where kinetic energy is higher than equilibrium, damping excess energy effectively. It reduces the diffusion density along axes where the system is under-energized, preserving fine structures and reducing over-smoothing. This adaptive behaviour is critical in guiding the diffusion process toward a more physical evolution.

Anisotropic 3D Gaussian Modeling. Unlike adding isotropic noise in DDPM (Ho, Jain, and Abbeel 2020), we add anisotropic 3D Gaussian noise per axis. Directions with excess kinetic energy $\beta_{t,x}^E$ get more noise decay, and under-energized directions get boosted, driving the system more precisely toward equilibrium. Hence, our novel approach mimics anisotropic thermalization processes of real physical systems. Figure 4 (b) illustrates the 3D Gaussian noises added for 3D point clouds. Adding noise per axis allows the model to reflect the variations of energy distribution per axis, which enhances generalization abilities for unseen datasets that are never trained.

To add anisotropic 3D Gaussian noise along the x, y and z axes, we reconstruct the $\beta_{t,i}^E$ into a diagonal matrix β_t^E, as defined by Equation 6.

$$\beta_t^E = diag(\beta_{t,x}^E, \beta_{t,y}^E, \beta_{t,z}^E) \tag{6}$$

The forward process is then defined by Equation 7.

$$\begin{aligned}
&q(X^{(0:T)}) = q(X^0) \prod_{t=1}^{T} q(X^t \mid X^{t-1}) \\
&q(X^t \mid X^{t-1}) = \mathcal{N}(X^t \mid \boldsymbol{\mu_q}(X^t, t), \boldsymbol{\Sigma_q}(X^t, t)) \\
&\boldsymbol{\mu_q} = diag\left(\sqrt{(1-\beta_{t,x}^E)}, \sqrt{(1-\beta_{t,y}^E)}, \sqrt{(1-\beta_{t,z}^E)}\right) \cdot X^{t-1} \\
&\boldsymbol{\Sigma_q} = diag\left(\beta_{t,x}^E, \beta_{t,y}^E, \beta_{t,z}^E\right)
\end{aligned} \tag{7}$$

where $q(X^{(0:T)})$ is a ground truth diffusion distribution, which is defined by gradually adding Gaussian noise to the complete 3D point clouds X. $q(X^0)$ denotes the complete 3D point cloud distribution. $q(X^t \mid X^{t-1})$ is the Markov diffusion kernel, which adds noise at time step t and models the distribution of X at the next time step $t+1$. $diag\left(\sqrt{(1-\beta_{t,x}^E)}, \sqrt{(1-\beta_{t,y}^E)}, \sqrt{(1-\beta_{t,z}^E)}\right)$ is a diagonal matrix, indicating the mean of the distribution. $diag\left(\beta_{t,x}^E, \beta_{t,y}^E, \beta_{t,z}^E\right)$ is also a diagonal matrix, indicating the covariance of the distribution. The detailed derivation of the forward process is provided in Appendix A.

Reverse Process

The reverse process of 3DDM mimics the inverse processes of anisotropic thermalization in real physical systems. It learns to denoise along per-axis and per-timestep anisotropically and adapts to the variability of x, y and z axes, as defined by Equation 8. The $p_\theta(X^{t-1} \mid X^t, c)$ in 3DDM is modeled as 3D Gaussian distributions with a covariance matrix Σ_θ, whose three variances in the diagonal are unequal. This implies that the network learns the anisotropic noise along x, y, z axes independently.

$$\begin{aligned}
&p_\theta\left(X^{t-1} \mid X^t, c\right) = \mathcal{N}\left(X^{t-1} \mid \boldsymbol{\mu_\theta}\left(X^t, t, c\right), \boldsymbol{\Sigma_\theta}\left(X^t, t, c\right)\right) \\
&\boldsymbol{\mu_\theta} = diag\left(\frac{1}{\sqrt{\alpha_{t,x}^E}}, \frac{1}{\sqrt{\alpha_{t,y}^E}} \frac{1}{\sqrt{\alpha_{t,z}^E}}\right) \cdot \\
&\left(X^t - diag\left(\frac{\beta_{t,x}^E}{\sqrt{1-\overline{\alpha}_{t,x}^E}}, \frac{\beta_{t,y}^E}{\sqrt{1-\overline{\alpha}_{t,y}^E}}, \frac{\beta_{t,z}^E}{\sqrt{1-\overline{\alpha}_{t,z}^E}}\right) \cdot \epsilon_\theta(X^t, t, c)\right) \\
&\boldsymbol{\Sigma_\theta} = diag\left(\frac{(1-\overline{\alpha}_{t-1,x}^E)\beta_{t,x}^E}{1-\overline{\alpha}_{t,x}^E}, \frac{(1-\overline{\alpha}_{t-1,y}^E)\beta_{t,y}^E}{1-\overline{\alpha}_{t,y}^E}, \frac{(1-\overline{\alpha}_{t-1,z}^E)\beta_{t,z}^E}{1-\overline{\alpha}_{t,z}^E}\right)
\end{aligned} \tag{8}$$

where θ is the network parameter. c denotes latent features of a partial 3D point cloud. $\boldsymbol{\mu_\theta}$ is the mean of predicted 3D Gaussian distributions. ϵ_θ is the noise predicted by the network. $\boldsymbol{\Sigma_\theta}$ is a diagonal matrix, indicating the covariance of the 3D Gaussian distributions.

To derive the reverse process $p_\theta(X^{t-1} \mid X^t, c)$, the probability density function of 3D Gaussian distribution is used to achieve $\boldsymbol{\mu_\theta}$ and $\boldsymbol{\Sigma_\theta}$. Refer to Appendix B for the detailed derivation of the reverse process.

Thus, X^{t-1} is sampling by $X^{t-1} = \boldsymbol{\mu_\theta} + \sqrt{\boldsymbol{\Sigma_\theta}} \cdot \boldsymbol{\epsilon_\theta}(X^t, t, c) + \sigma_t \epsilon$, where $\sigma_t = \sqrt{\beta_t^E}$ and $\epsilon \sim \mathcal{N}(0, I)$.

Anisotropic Quadratic Loss Function

Unlike the MSE loss in standard DDPM, we propose a novel anisotropic quadratic loss for 3DDM, which is derived by maximizing the evidence lower bound of the 3D point sets.

Methods	34 seen categories						21 unseen categories					
	F1-S ↑	F1-M ↑	F1-H ↑	CD-S ↓	CD-M ↓	CD-H ↓	F1-S ↑	F1-M ↑	F1-H ↑	CD-S ↓	CD-M ↓	CD-H ↓
PCN (Yuan et al. 2018)	0.19	0.19	0.17	2.15	2.21	2.52	0.12	0.11	0.10	2.80	2.92	3.43
FoldingNet (Yang et al. 2018)	0.16	0.15	0.12	2.21	2.28	2.60	0.10	0.09	0.07	2.66	2.79	3.24
TopNet (Tchapmi et al. 2019)	0.11	0.10	0.08	2.78	2.89	3.37	0.07	0.06	0.04	3.44	3.60	4.23
AnchorFormer (Chen et al. 2023)	0.30	0.30	0.27	1.44	1.48	**1.76**	0.26	0.25	0.22	**1.61**	1.71	2.67
SeedFormer (Zhou et al. 2022)	0.10	0.09	0.08	2.30	2.53	3.10	0.07	0.07	0.06	2.57	2.85	3.65
FSC (Wu et al. 2024)	0.21	0.20	0.18	1.97	2.03	2.33	0.13	0.12	0.11	2.49	2.60	3.12
3DDM (Ours)	**0.41**	**0.42**	**0.38**	**1.42**	**1.48**	1.90	**0.37**	**0.36**	**0.31**	1.72	**1.69**	**2.37**

Table 1: Evaluations of 3D point cloud completion on 34 seen and 21 unseen categories. -S, -M and -H denote the simple, median and hard modes with the missing rate of 25%, 50% and 75%, respectively. (Higher F1 and lower CD are better.)

$$\log p_\theta\left(X^0|c\right) \geq \mathbb{E}_{q\left(X^{(1:T)}|X^0\right)}\left[\log \frac{p_\theta\left(X^{(0:T)}|c\right)}{q\left(X^{(1:T)} \mid X^0\right)}\right] =$$
$$\underbrace{KL\left(q\left(X^T \mid X^0\right) \| p_\theta\left(X^T\right)\right)}_{\text{① prior matching term}} - \underbrace{E_{q\left(X^1|X^0\right)}\left[\log p_\theta\left(X^0 \mid X^1, c\right)\right]}_{\text{② reconstruction term}}$$
$$+ \underbrace{\sum_{t-2}^{T} E_{q\left(X^t|X^0\right)}\left[KL\left(q\left(X^{t-1} \mid X^t, X^0\right) \| p_\theta\left(X^{t-1} \mid X^t, c\right)\right)\right]}_{\text{③ denoising matching term}} \tag{9}$$

① has no trainable parameters and is equal to zero.

③ is a denoising matching term. The $q\left(X^{t-1} \mid X^t, X^0\right)$ transition step acts as a ground-truth signal, since it defines how to denoise a noisy 3D point cloud anisotropically in the forward process of 3DDM. The 3DDM learns the desired denoising transition step $p_\theta\left(X^{t-1} \mid X^t, c\right)$ as an approximation to tractable, ground-truth denoising transition step $q\left(X^{t-1} \mid X^t, X^0\right)$. The term ③ is therefore minimized when the two denoising steps match as closely as possible, as measured by their KL Divergence.

② can be interpreted as a 3D point cloud reconstruction term. Substituting ② into ③, the final term is :

$$\sum_{t-1}^{T} E_{q\left(X^t|X^0\right)}\left[KL\left(q\left(X^{t-1} \mid X^t, X^0\right) \| p_\theta\left(X^{t-1} \mid X^t, c\right)\right)\right] \tag{10}$$

Since $q\left(X^{t-1} \mid X^t, X^0\right)$ and $p_\theta\left(X^{t-1} \mid X^t, c\right)$ are modeled as the 3D Gaussian distributions, the KL Divergence between two 3D Gaussian distributions in ③ is:

$$KL(\mathcal{N}(X^{t-1}; \boldsymbol{\mu_q}, \boldsymbol{\Sigma_q}) || \mathcal{N}(X^{t-1}; \boldsymbol{\mu_\theta}, \boldsymbol{\Sigma_\theta})) = \frac{1}{2} \cdot$$
$$\left[Tr(\boldsymbol{\Sigma_\theta}^{-1}\boldsymbol{\Sigma_q}) + (\boldsymbol{\mu_\theta} - \boldsymbol{\mu_q})^T \boldsymbol{\Sigma_\theta}^{-1}(\boldsymbol{\mu_\theta} - \boldsymbol{\mu_q}) + ln\left(\frac{|\boldsymbol{\Sigma_\theta}|}{|\boldsymbol{\Sigma_q}|}\right) - d\right] \tag{11}$$

where $Tr(\cdot)$ denotes the trace of a matrix. Σ_θ^{-1} is the inverse covariance matrix of p_θ. $d = 3$ for 3D Gaussians.

Furthermore, Equation 11 simplifies to:

$$L_{AQ} = (\boldsymbol{\mu_\theta} - \boldsymbol{\mu_q})^T \boldsymbol{\Sigma_\theta}^{-1}(\boldsymbol{\mu_\theta} - \boldsymbol{\mu_q}) \tag{12}$$

Substituting Equations 7 and 8 into Equation 12, the training loss function for the 3DDM is:

$$L_{AQ} = (\boldsymbol{\epsilon_t} - \boldsymbol{\epsilon_\theta})^T \cdot \boldsymbol{A} \cdot (\boldsymbol{\epsilon_t} - \boldsymbol{\epsilon_\theta}), \quad \boldsymbol{A} =$$
$$diag\left(\frac{\beta^E_{t,x}}{(1-\beta^E_{t,x}) \cdot \bar{\alpha}^E_{t,x}}, \frac{\beta^E_{t,y}}{(1-\beta^E_{t,y}) \cdot \bar{\alpha}^E_{t,y}}, \frac{\beta^E_{t,z}}{(1-\beta^E_{t,z}) \cdot \bar{\alpha}^E_{t,z}}\right) \tag{13}$$

where $\boldsymbol{A}$ introduces direction-dependent weighting, meaning errors along certain axes may be penalized more heavily than others depending on $\boldsymbol{A}$. $\boldsymbol{A}$ is symmetric positive definite, which is exactly anisotropic.

Experiments

Our 3DDM was trained on an Nvidia GeForce 2080Ti GPU with 12 GB memory using PyTorch. Implementation details are provided in Appendix D.

Datasets

We use the ShapeNet34 (Yu et al. 2021) train/test splits (46,765 for training and 5,750 for testing) from 34 categories, as it has been done in previous methods (Chen et al. 2023), and (Wu et al. 2024).

Unseen evaluation datasets are never trained and are only used for evaluating the generalization capability of each network. We select three unseen datasets, including 3D objects and indoor and outdoor scenes, which are captured by various sensors with different densities and varied point distributions, posing significant challenges to all networks. The three unseen datasets are: ShapeNet21 (Yu et al. 2021) for 3D objects, ScanNet (Dai et al. 2017) for 3D real-world indoor scenes and KITTI (Geiger, Lenz, and Urtasun 2012) for 3D real-world outdoor scenes. See Appendix E for details.

Evaluation Metrics

To evaluate partial 3D point clouds with ground-truth, we use Chamfer Distance (CD) (Fan, Su, and Guibas 2017) and F1 Score (Knapitsch et al. 2017; Tatarchenko et al. 2019) as the evaluation metrics following PoinTr (Yu et al. 2021) and FSC (Wu et al. 2024).

To evaluate the partial 3D point clouds without the ground truth of complete 3D point clouds, we use Minimal Matching Distance (MMD) and Fidelity as the evaluation matrices following PCN (Yuan et al. 2018) and AnchorFormer (Chen et al. 2023). MMD finds the minimum CD between completion results and the ShapeNet dataset. Fidelity is the average CD, which measures how well the partial input is preserved.

Evaluation on ShapeNet34 and unseen ShapeNet21

To evaluate the completion performance, all networks are trained and evaluated on the ShapeNet34 dataset. To further

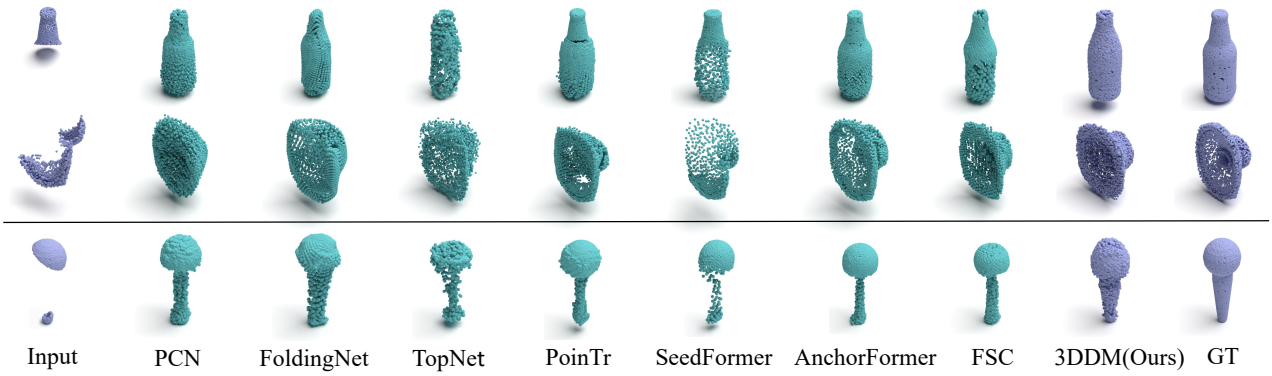


Figure 5: Qualitative completion results on the ShapeNet34 (top) and unseen ShapeNet21 (bottom).

Methods		PCN	FoldingNet	TopNet	PoinTr	PVD	AnchorFormer	SeedFormer	FSC	3DDM (Ours)
ScanNet Chair	MMD ↓	5.11	9.55	4.83	8.11	5.08	6.54	6.08	4.32	**4.04**
	Fidelity ↓	0.05	0.97	0.97	-	1.76	0.02	0.04	0.04	**0.03**
ScanNet Table	MMD ↓	4.62	6.93	4.99	7.49	4.55	6.04	5.08	4.22	**4.20**
	Fidelity ↓	0.80	0.56	2.40	-	3.35	0.52	0.40	2.85	**0.01**
KITTI	MMD ↓	1.84	1.69	1.79	2.18	4.10	3.00	2.87	1.25	**0.07**
	Fidelity ↓	0.18	0.22	0.19	-	0.29	0.16	0.03	0.21	**0.03**

Table 2: Evaluations on the unseen real-world ScanNet and KITTI datasets that were not trained. MMD and Fidelity are reported multiplied by 10^2. Lower MMD and Fidelity are better. '-' denotes that PoinTr has a severe problem on the scales of completion results for unseen real-world scenes.

evaluate the generalization ability on unseen categories with different shape patterns that were not trained, we evaluate these trained models on the untrained ShapeNet21 dataset.

The average F1 and CD on all 34 categories are reported in Table 1 (left) with qualitative results in Figure 5 (top). 3DDM achieves state-of-the-art scores on the whole ShapeNet34 testing dataset. Table 1 (right) reports the average F1 and CD on all unseen 21 categories with qualitative results in Figure 5 (bottom). Our 3DDM outperforms other state-of-the-art methods and shows a strong generalization ability for 21 unseen categories that were never trained. AnchorFormer is followed by our method and still shows great performance in unseen categories. Please refer to Appendix F for more detailed results.

Evaluation on Unseen Real-World ScanNet Scenes

We evaluate the generalization ability of all networks on real-world semantic ScanNet indoor scenes (Dai et al. 2017). All methods are trained solely on ShapeNet34 and directly evaluated on ScanNet. Compared to synthetic datasets in the training dataset, the point clouds in ScanNet are significantly noisier and exhibit large missing data, posing a more challenging cross-domain completion task.

As shown in Figure 6 (top), our 3DDM successfully reconstructs dense and structurally complete geometries, particularly excelling in regions such as floors and chair legs. In contrast, PoinTr, AnchorFormer and FSC frequently suffer from poor completion performance in missing structures (e.g. chair legs). The quantitative results are reported in Table 2. Our 3DDM achieves top completion performance in both MMD and Fidelity, indicating that 3DDM achieves superior completion accuracy and strong cross-domain generalization capabilities on unseen real-world scenes. More qualitative comparisons are provided in Appendix G.

Evaluation on Unseen Real-World KITTI Scenes

We further evaluate the generalization ability of all networks on LiDAR KITTI cars (Geiger, Lenz, and Urtasun 2012). All networks are trained on ShapeNet34 and evaluated on KITTI cars. Unlike data in the training data, 3D point clouds from LiDAR scans are very sparse and incomplete, with different point distributions. The 1,000 KITTI cars contain 400 points on average, with some having fewer than 5 points.

As shown in Table 2, our 3DDM achieves the best in both MMD and Fidelity, indicating strong cross-domain generalization ability on unseen LiDAR scans. Figure 6 (bottom) visualizes comparisons across different methods, and Figure 7 shows details of completion results. Existing methods struggle with sparse and noisy KITTI cars. For instance, PoinTr significantly underestimates the car scales and sizes for car number 1 and car number 2, as shown in Figure 7 (third row). AnchorFormer fails to generate the missing points in the frontal parts (fifth row in Figure 7). In contrast, our 3DDM generates more faithful and more accurate results in both size and geometry, which highlights the practical potential of 3DDM in safety-critical applications such as autonomous driving, where reliable 3D shape completion is essential. Refer to Appendix H for additional comparisons.

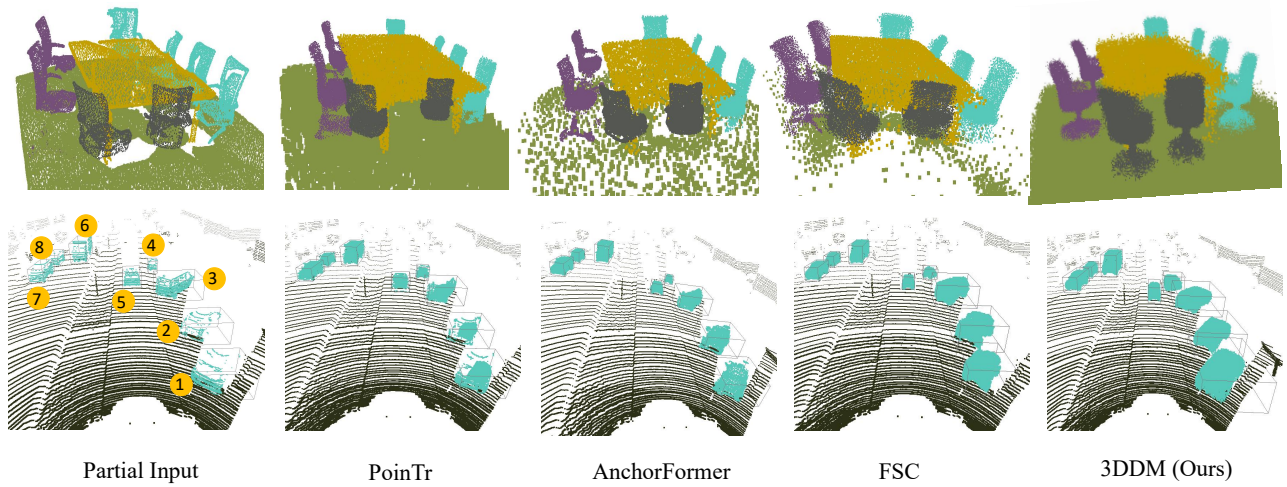


Figure 6: Qualitative completion results on the unseen real-world ScanNet (top) and KITTI (bottom) that were not trained.

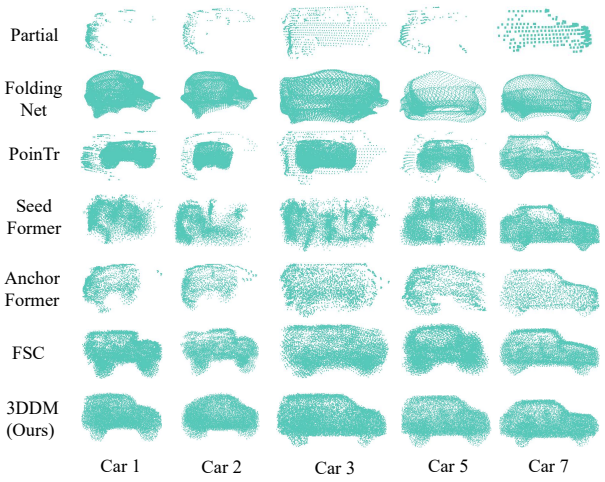


Figure 7: Details of KITTI cars in Figure 6.

Methods	F1-S / -M / -H ↑	CD-S / -M / -H ↓	Parameters
PointNet	0.82 / 0.81 / 0.73	0.68 / 0.70 / 0.89	1,577,886
PointNet++	0.65 / 0.63 / 0.58	1.26 / 1.34 / 1.47	1,740,894
PCN	0.82 / 0.80 / 0.72	0.70 / 0.70 / 0.92	1,426,974
DGCNN	0.83 / 0.82 / 0.73	0.64 / 0.66 / 0.86	1,176,222
PVCNN	0.79 / 0.76 / 0.68	0.76 / 0.81 / 1.03	2,708,062

Table 3: Feature extraction analysis.

Steps	F1-S ↑ / F1-M ↑ / F1-H ↑	CD-S ↓ / CD-M ↓ / CD-H ↓
1	0.31 / 0.30 / 0.25	7.89 / 8.08 / 8.42
5	0.53 / 0.53 / 0.47	1.63 / 1.69 / 1.88
10	0.58 / 0.57 / 0.51	1.15 / 1.21 / 1.45
20	**0.60 / 0.59 / 0.54**	**1.05 / 1.09 / 1.35**
50	0.60 / 0.58 / 0.52	1.04 / 1.09 / 1.33
100	0.61 / 0.60 / 0.54	1.01 / 1.07 / 1.33
500	0.57 / 0.56 / 0.50	1.10 / 1.26 / 1.27
800	0.55 / 0.55 / 0.50	1.20 / 1.28 / 1.31

Table 4: Diffusion steps analysis.

Ablation Studies

Feature extraction analysis We conduct an ablation study using five state-of-the-art encoders in PointNet (Qi et al. 2017a), PointNet++ (Qi et al. 2017b), PCN (Yuan et al. 2018), DGCNN (Wang et al. 2019), and PVCNN (Liu et al. 2019). The comparison is performed on the guitar category. The results are reported in Table 3. As can be seen, all five encoders can be adopted in our 3DDM for feature extraction, and DGCNN achieves the best completion performance.

Diffusion steps analysis To analyze the impact of diffusion steps on the completion performance of our 3DDM, we train the 3DDM with varying diffusion steps. As reported in Table 4, the 3DDM still achieves strong completion ability with fewer than 20 diffusion steps, which shows the effectiveness of the physically-based diffusion process in 3DDM. 3DDM achieves the best performance with 100 steps. More detailed ablation analyses are provided in the Appendix.

Conclusion

This paper has presented a physically-based anisotropic and energy-aware diffusion model, 3DDM, for 3D point cloud completion. By incorporating anisotropic diffusion based on kinetic energy evolution, energy-guided noise modulation, and anisotropic denoising with a tailored loss function, 3DDM effectively simulates the non-equilibrium dynamics and anisotropic behaviors of real-world 3D particle systems, enhancing the physical plausibility and structural integrity of the generated point clouds. Experiments demonstrate that 3DDM achieves strong performance across multiple benchmarks, with superior generalization to unseen categories and real-world scenes, while significantly improving sampling efficiency. 3DDM offers a new perspective for building efficient and physically consistent and realistic 3D generative models.

Acknowledgments

This work was supported by the Natural Science Foundation of Shaanxi Province, China under Grant No. 2025JC-YBQN-940.

References

Chen, X.; Shi, S.; Ma, T.; Zhou, J.; See, S.; Cheung, K. C.; and Li, H. 2025. M3net: Multimodal multi-task learning for 3d detection, segmentation, and occupancy prediction in autonomous driving. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 2275–2283.

Chen, Z.; Long, F.; Qiu, Z.; Yao, T.; Zhou, W.; Luo, J.; and Mei, T. 2023. Anchorformer: Point cloud completion from discriminative nodes. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13581–13590.

Cheng, Y.; Lee, H.; Tulyakov, S.; Schwing, A. G.; and Gui, L. 2023. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4456–4465.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5828–5839.

Fan, H.; Su, H.; and Guibas, L. J. 2017. A point set generation network for 3d object reconstruction from a single image. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 605–613.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Giovanni, B.; and Michele, P. 2007. Accurate sampling using Langevin dynamics. *Physical Review E*, 75: 056707.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27.

Gu, Z.; Ma, J.; Huang, Y.; Wei, H.; Chen, Z.; Zhang, H.; and Hong, W. 2025. HGSFusion: Radar-camera fusion with hybrid generation and synchronization for 3d object detection. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3185–3193.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Ji, Y.; He, B.; Tan, Z.; and Wu, L. 2025. Game4loc: A uav geo-localization benchmark from game data. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3913–3921.

Knapitsch, A.; Park, J.; Zhou, Q.-Y.; and Koltun, V. 2017. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4): 1–13.

Liu, Z.; Tang, H.; Lin, Y.; and Han, S. 2019. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32.

Novotny, J.; and Laidlaw, D. H. 2024. ting Text ReEvaluaading Speed in VR Scenes and 3D Particle Visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 30(5): 2602–2612.

Pan, S.; Xu, Y.; Xu, R.; Zhou, Z.; Wu, S.; and Yu, Z. 2025. Self-Correcting Robot Manipulation via Gaussian-Splatted Foresight. In *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 26642–26650.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Advances in Neural Information Processing Systems*, NIPS'17, 5105–5114. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Tatarchenko, M.; Richter, S. R.; Ranftl, R.; Li, Z.; Koltun, V.; and Brox, T. 2019. What do single-view 3d reconstruction networks learn? In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3405–3414.

Tchapmi, L. P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; and Savarese, S. 2019. Topnet: Structural point cloud decoder. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 383–392.

Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.; Bronstein, M.; and Solomon, J. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38: 1–12.

Wu, X.; Wu, X.; Luan, T.; Bai, Y.; Lai, Z.; and Yuan, J. 2024. FSC: Few-point Shape Completion. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26077–26087.

Yang, Y.; Feng, C.; Shen, Y.; and Tian, D. 2018. FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Yu, X.; Rao, Y.; Wang, Z.; Liu, Z.; Lu, J.; and Zhou, J. 2021. Pointr: Diverse point cloud completion with geometry-aware transformers. In *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12498–12507.

Yuan, W.; Khot, T.; Held, D.; Mertz, C.; and Hebert, M. 2018. Pcn: Point completion network. In *In International Conference on 3D Vision*, 728–737. IEEE.

Zhou, H.; Cao, Y.; Chu, W.; Zhu, J.; Lu, T.; Tai, Y.; and Wang, C. 2022. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *In European Conference on Computer Vision*, 416–432. Springer.

Zhou, L.; Du, Y.; and Wu, J. 2021. 3d shape generation and completion through point-voxel diffusion. In *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5826–5835.