Bayesian Diffusion Models for 3D Shape Reconstruction

Haiyang Xu[*,1] Yu Lei[*,2] Zeyuan Chen[3] Xiang Zhang[3]
Yue Zhao[4] Yilin Wang[4] Zhuowen Tu[3]
[1] University of Science and Technology of China [2] Shanghai Jiao Tong University
[3] University of California, San Diego [4] Tsinghua University

Abstract

We present Bayesian Diffusion Models (BDM), a prediction algorithm that performs effective Bayesian inference by tightly coupling the top-down (prior) information with the bottom-up (data-driven) procedure via joint diffusion processes. We show the effectiveness of BDM on the 3D shape reconstruction task. Compared to prototypical deep learning data-driven approaches trained on paired (supervised) data-labels (e.g. image-point clouds) datasets, our BDM brings in rich prior information from standalone labels (e.g. point clouds) to improve the bottom-up 3D reconstruction. As opposed to the standard Bayesian frameworks where explicit prior and likelihood are required for the inference, BDM performs seamless information fusion via coupled diffusion processes with learned gradient computation networks. The specialty of our BDM lies in its capability to engage the active and effective information exchange and fusion of the top-down and bottom-up processes where each itself is a diffusion process. We demonstrate state-of-the-art results on both synthetic and real-world benchmarks for 3D shape reconstruction. Project link: https://mlpc-ucsd.github.io/BDM

1. Introduction

The Bayesian theory [5, 38], under the general principle of *analysis-by-synthesis* [80], has made a profound impact on a myriad of tasks in computer vision and machine learning, including face modeling [13], shape detection and tracking [6], image segmentation [65], scene categorization [18], image parsing [66], depth estimation [43, 73], object recognition [20, 21, 70], and topic modeling [7].

We assume the task of predicting $\mathbf{y}$ for a given input $\mathbf{x}$ ($\mathbf{y}$ and $\mathbf{x}$ represent respectively the 3D point clouds and the input image in this paper). The Bayes' theorem turns the posterior $p(\mathbf{y}|\mathbf{x})$ into the product of the likelihood $p(\mathbf{x}|\mathbf{y})$ and the prior $p(\mathbf{y})$ as $p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$, which can be further approximated by $p_\gamma(\mathbf{y}|\mathbf{x})p(\mathbf{y})$ [40], where $p_\gamma(\mathbf{y}|\mathbf{x})$ represents a direct bottom-up (data-driven) process, *e.g.*, the Viola-Jones face detector [68]. The formulations of $p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$ [41] and $p_\gamma(\mathbf{y}|\mathbf{x})p(\mathbf{y})$ [40] can be solved via *e.g.*, the Markov Chain Monte Carlo (MCMC) sampling methods [3, 71].

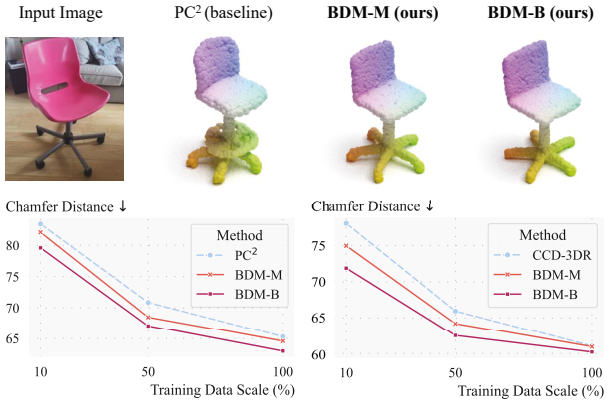

Figure 1. **Baseline vs Bayesian Diffusion Models**. Our BDM brings rich prior knowledge into the shape reconstruction process, fixing the incorrect predictions by the baseline (top row). BDM surpasses baselines in all three training data scales (bottom row).

When does the Bayesian help? The Bayesian theory [5, 34, 80] provides a principled statistical foundation to guide the top-down and bottom-up inference, which is deemed to be of great biological significance [38]. In the early development of computer vision [50], the objects in study [13, 21, 70] are often of simplicity and are picked from relatively small-scale datasets [19, 27, 54]. The top-down prior [20, 66] can therefore provide a strong regularization and inductive bias to the bottom-up process that has been trained from the data [19, 27].

Why is the Bayesian not anymore widely adopted in the deep learning era? Although still being an active subject [22] in study, the top-down/prior information has not been widely adopted in the big-data/deep-learning era, where an immediate improvement can be shown over the data-driven models [16, 39, 58] learned from large-scale training set of input and ground-truth pairs $S_s = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1..n\}$. The

* equal contribution. Work done during the internship of Haiyang Xu, Yu Lei, Yue Zhao, and Yilin Wang at UC San Diego.

reasons are threefold: **1)** *Rich features* from large-scale data [14] become substantially more robust than manually designed ones [47], whereas the top-down prior $p(\mathbf{y})$ for structured output $\mathbf{y}$ no longer shows an improvement. **2)** *Strong bottom-up models* [16, 39, 58] learned in a $\mathbf{x} \rightarrow \mathbf{y}$ fashion from paired/supervised dataset, $S_s = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1..n\}$, are powerful, and they do not necessarily see an immediate benefit from introducing a separate prior $p(\mathbf{y})$ that is obtained from $S_l = \{\mathbf{y}_i, i = 1..n\}$ alone, as knowledge about the $\mathbf{y}$ has already been implicitly captured in the data-driven $p_\gamma(\mathbf{y}|\mathbf{x})$. **3)** The presence of the intermediate stages with different architectural designs for the deep models makes merely combining the data-driven model $p_\gamma(\mathbf{y}|\mathbf{x})$ and the prior $p(\mathbf{y})$ not so obvious, as the distributions for $p_\gamma(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{y})$ are hard to model and may not co-exist.

The emerging opportunity for combining bottom-up and top-down processes with diffusion-based models. The recent development in diffusion models [31, 59, 61, 62] has led to substantial improvements to unsupervised learning beyond the traditional VAE [37] and adversarial learning [26, 33, 64]. The presence of diffusion models for learning both $p(\mathbf{y})$ (*e.g.* shape priors [84]) and $p_\gamma(\mathbf{y}|\mathbf{x})$ (*e.g.* 3D shape reconstruction [15, 51]) inspires us to develop a new inference algorithm, Bayesian Diffusion Models, that is applied to single-view 3D shape reconstruction.

The contribution of our paper is summarized as follows:

- We present **Bayesian Diffusion Models** (BDM), a new statistical inference algorithm that couples diffusion-based bottom-up and top-down processes in a joint framework. BDM is particularly effective when having separately available data-labeling (supervised) dataset $S_s = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1..n\}$ and standalone label dataset $S_l = \{\mathbf{y}_i, i = 1..m\}$ for training $p_\gamma(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{y})$ respectively. For example, obtaining a set [63] for real-world images with the corresponding ground-truth 3D shapes is challenging whereas a large dataset of standalone 3D object shapes such as ShapeNet [9] is readily available.
- Two strategies for fusing the information exchange between the bottom-up and the top-down diffusion process are developed: 1) a **blending procedure** that takes the two processes in a plug-in-and-play fashion, and 2) a **merging procedure** that is trained.
- We emphasize the key property of **fusion-with-diffusion** in BDM vs. **fusion-by-combination** in the traditional MCMC Bayesian inference. BDM also differs from the current pre-training + fine-tuning process [8] and the prompt engineering practice [30] by making the bottom-up and top-down integration process transparent and explicit; BDM points to a promising direction in computer vision and machine learning with a new diffusion-based Bayesian method.

BDM demonstrates the state-of-the-art results on the single image 3D shape reconstruction benchmarks.

2. Related Work

Bayesian Inference. As stated previously, the Bayesian theory [5, 38] has been adopted in a wide range of computer applications[6, 13, 18, 20, 21, 43, 66, 70, 73], but the results of these approaches are less competitive than those by the deep learning based ones [28, 39, 58].

3D Shape Generation. Early 3D shape generation methods [1, 24, 29, 55, 74, 79] typically leverage variational auto-encoders (VAE) [37] and generative adversarial networks (GAN) [26] to learn the distribution of the 3D shape. Recently, the superior performance of diffusion models in generative tasks also makes them the go-to methods in 3D shape generation. PVD [84] proposes to diffuse and denoise on point clouds using a Point-Voxel-CNN [46], while in DMPGen [48], the diffusion process is modeled by a PointNet [57]. LION [81] uses a hierarchical VAE to encode 3D shapes into latents where the diffusion and generative processes are performed. Another popular category of 3D generative models leverages 2D text-to-image diffusion models as priors and lifts them to 3D representations [42, 53, 56, 69].

Single-View 3D Reconstruction. Recovering 3D object shapes from a single view is an ill-posed problem in computer vision. Traditional approaches extract multi-modal information, including shading [4, 32], texture [72], and silhouettes [10], for reconstructing 3D shapes. Learning-based reconstruction methods become popular with the advance of neural networks and the availability of large-scale 2D-3D datasets [9, 23]. In these methods, different 3D representations are employed, including voxel grids [12, 25, 76, 77, 83], point clouds [17, 49], meshes [35, 36, 75], and implicit functions [11, 52, 60].

Some other reconstruction methods take images as conditions for generative models [15, 51, 74]. In particular, they learn the prior shape distribution from large-scale 3D datasets and then perform 3D reconstruction with 2D image observations. 3D-VAE-GAN [74] employs GAN and VAE to learn a generator mapping from a low-dimensional probabilistic space to a 3D-shape space, and feeds images to the generator for reconstruction. Recent methods for single-view reconstruction are mostly based on diffusion models [15, 44, 45, 51]. PC^2 [51] proposes to project encoded features from 2D back to 3D, which facilitates the point cloud reconstruction in denoising. CCD-3DR [15] introduces a centered diffusion probabilistic model based on PC^2, which offers better consistency in alignments of local features and final prediction results. RenderDiffusion [2] presents an explicit latent 3D representation into a diffusion model, yielding a 3D-aware pipeline that could perform 3D reconstruction. Zero1-to-3 [45] and One-2-3-45 [44] inject camera information into a 2D diffusion model and reconstruct 3D shapes with synthesized multi-view images.

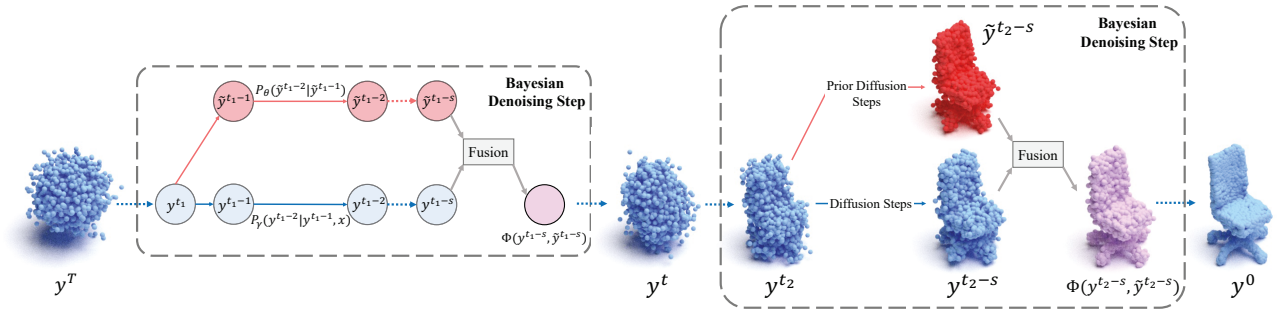

Figure 2. Overview of the generative process in our Bayesian Diffusion Model. In each Bayesian denoising dtep, the prior diffusion model fuses with the reconstruction process, bringing rich prior knowledge and improving the quality of the reconstructed point cloud. We illustrate our Bayesian denoising step in two ways, left in the form of a flowchart and right in the form of point clouds.

Prompts and Latent Representations in Transformers. Transformers [67] provide a general tokenized learning framework, allowing the insertion of the representation of $\mathbf{y}$ into $p_\gamma(\mathbf{y}|\mathbf{x})$ as special tokens [10, 30]. However, the prior knowledge in Transformers serves as a latent condition that is often opaque and non-interpretable.

3. Method

In the following section, we introduce the Bayesian Diffusion Models, the framework of which is illustrated in Fig. 2. Initially, a concise overview of denoising diffusion models, particularly focusing on point cloud diffusion models, is presented. This is followed by an exposition of its conditional variant applied to 3D shape reconstruction. Subsequently, we delve into the central concept of our proposed Bayesian Diffusion Models which integrate Bayesian priors. Concluding this section, we provide an in-depth exploration of our novel prior-integration methodology.

3.1. Bayesian Inference with Stochastic Gradient Langevin Dynamics

As discussed in Sec. 1, our task is to predict $\mathbf{y}$ (a set of point clouds) for a given input $\mathbf{x} \in \mathbb{R}^q$ (an input image). For the 3D shape reconstruction task, the output $\mathbf{y}$ consists of a set of 3D points. The Bayesian theory [5] focuses on the study of the posterior $p(\mathbf{y}|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$, where $p(\mathbf{y})$ is the prior (top-down information). The likelihood term $p(\mathbf{x}|\mathbf{y})$ can be alternatively replaced [40] by a data-driven (bottom-up) distribution $p_\gamma(\mathbf{y}|\mathbf{x})$, which is learned from a paired data-labels training set $S_s = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1..n\}$. Therefore, the inference for the optimal prediction $\mathbf{y}^*$ can then be carried via Markov Chain Monte Carlo (MCMC) [3] using stochastic gradient Langevin dynamics [71]:

$$\Delta \mathbf{y}^t = \frac{\epsilon^t}{2}\Big(\underbrace{\nabla \log p_\gamma(\mathbf{y}^t|\mathbf{x})}_{data-driven} + \underbrace{\nabla \log p(\mathbf{y}^t)}_{prior}\Big) + \eta^t$$
$$\eta^t \sim N(0, \epsilon^t) \tag{1}$$

where ϵ^t denotes a sequence of step size and η^t is a Gaussian noise. The challenge in implementing Equation 1 is twofold: **1)** it requires knowing the explicit formulation for both the $\log p_\gamma(\mathbf{y}^t|\mathbf{x})$ and $\log p(\mathbf{y}^t)$, which is hard to obtain in real-world applications. **2)** a mere summation for the gradients $\nabla \log p_\gamma(\mathbf{y}|\mathbf{x})$ and $\nabla \log p(\mathbf{y}^t)$ limits the level of interaction between the bottom-up and top-down processes. Fig. 3 shows the basic Bayes formulation and demonstrates the stochastic gradient Langevin inference result.

3.2. Denoising Diffusion Probabilistic Models

The denoising diffusion models [31, 61, 62] have demonstrated superior performance in representing the structured data of high-dimension in both paired data-labels setting (learning $\nabla \log p^t(\mathbf{y})$ from $S_l = \{\mathbf{y}_i, i = 1..m\}$ [84]) and standalone labels settings (learning $\nabla \log p^t_\gamma(\mathbf{y}|\mathbf{x})$ from $S_s = \{(\mathbf{x}_i, \mathbf{y}_i), i = 1..n\}$ [51]) manners.

Here, we discuss the general formulation of the DDPM model [31, 61] in the context of single image 3D shape reconstruction. For point cloud generation, diffusion models are adept at learning the 3D shape of objects represented in point cloud format. At a high level, diffusion models iteratively denoise 3D points from a Gaussian sphere into a recognizable object. Consider a set of point clouds $\mathbf{y}^0$, consisting of N points, as an object in a $3N$-dimensional space. The diffusion model is employed to learn the mapping $s_\theta : \mathbb{R}^{3N} \to \mathbb{R}^{3N}$. Specifically, it is designed to estimate $q(\mathbf{y}^{t-1}|\mathbf{y}^t)$, which is the offset of the points from its position at timestep t. This approach aligns with standard diffusion model practices, which are trained to predict the noise $\epsilon \in \mathbb{R}^{3N}$ using the following loss function:

$$\mathcal{L} = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbf{I})}\left[\left\|\epsilon - s_\theta(\mathbf{y}^t, t)\right\|_2^2\right]. \tag{2}$$

The process of single-view reconstruction can be considered as a conditional point cloud generation. Diffusion models can be effectively applied here as well. Given a single-view image $\mathbf{x}$, the diffusion model aims to estimate $q(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x})$. Therefore, we should sample from

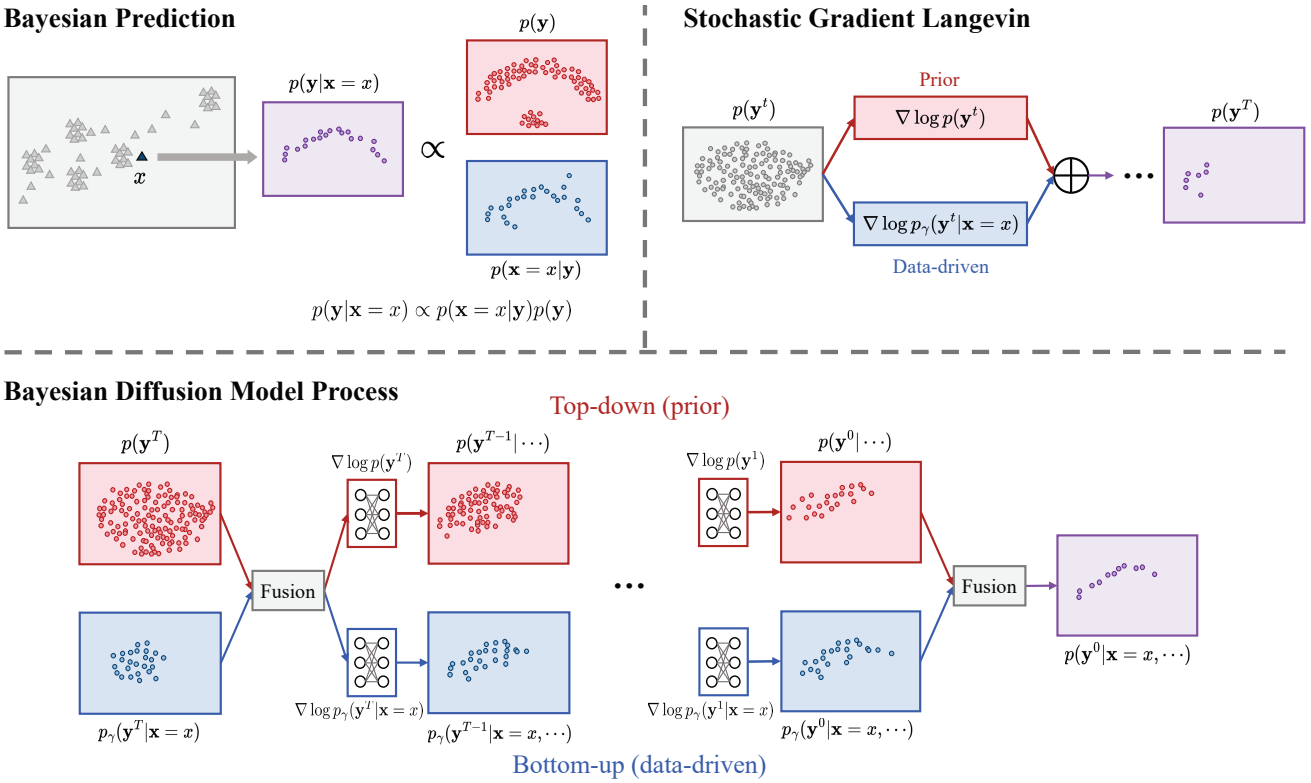

Figure 3. Illustration for the Bayesian Diffusion Models compared with the standard Bayesian formulation. We present the standard Bayesian formulation and the one using stochastic gradient Langevin on the top part, while our proposed BDM on the bottom.

$q(\mathbf{y}_{t-1}|\mathbf{y}_t, \mathbf{x})$ for this task. All the other concepts and principles are the same as point cloud generation.

3.3. Bayesian Diffusion Model

In our work, we denote $\mathbf{x}$ as an input image instance, while $\mathbf{y}^t, t = 1 \cdots T$ to represent the 3D object we want to reconstruct. γ is taken as the distribution learned by the reconstruction model and Π represents the distribution after fusing prior and γ.

Generative model: The shapes generated by prior model can be formulated as $p\left(\mathbf{y}^0\right)$. The noise predicted by the model at every timestep drives the spatial distribution according to the Markov Chain state transition probability $p\left(\mathbf{y}^t \mid \mathbf{y}^{t+1}\right)$. For the gradient, we have:

$$\varepsilon\left(\mathbf{y}^t\right) = -\sigma_t \nabla_{\mathbf{y}^t} \log p(\mathbf{y}^t) \tag{3}$$

Reconstruction model: In the same way of understanding generative diffusion model, the shapes generated by γ can be formulated as $p_\gamma\left(\mathbf{y}^0 \mid \mathbf{x}\right)$. The model can reconstruct point clouds from learned $p_\gamma\left(\mathbf{y}^t \mid \mathbf{y}^{t+1}, \mathbf{x}\right)$. In a similar way to the above, we have:

$$\varepsilon_\gamma\left(\mathbf{y}^t, \mathbf{x}\right) = -\sigma_t \nabla_{\mathbf{y}^t} \log p_\gamma(\mathbf{y}^t \mid \mathbf{x}) \tag{4}$$

It is obvious to conclude that the state transition probability of our Bayesian Diffusion Model comes from the corresponding probability from model γ and posterior from the prior model. As show in Fig. 2, we use an inaccessible function $\mathbf{\Phi}$ to incorporate the prior diffusion gradient into the reconstruction diffusion gradient. Accordingly, we can deduce the fused gradient in our Bayesian Diffusion Model as follows:

$$\begin{aligned}\nabla \log p_\pi(\mathbf{y}^t \mid \mathbf{x}) = \nabla_\gamma \log \Phi\big(&p_\gamma(\mathbf{y}^t \mid \mathbf{x}, \tilde{\mathbf{y}}^{t+1} \sim p_\pi(y^{t+1} \mid \mathbf{x}), \\ &p(\mathbf{y}^t \mid \tilde{\mathbf{y}}^{t+1} \sim p_\pi(\mathbf{y}^{t+1} \mid \mathbf{x})\big)\end{aligned} \tag{5}$$

3.4. Point Cloud Prior Integration

As stated in Sec. 1, diffusion-based models have provided us with the possibility to benefit from both bottom-up and top-down processes. Specifically, the multi step inference procedure allows more flexible and effective forms of function $\mathbf{\Phi}$ in Eq. (5). Therefore, we employ step-wise interaction between a generative diffusion model and a reconstruction model, facilitating closer integration of point cloud priors. In particular, we feed the intermediate point cloud from our reconstruction model into the prior model, forward it through a certain number of timesteps in both models and fuse two new point clouds from the prior model and the reconstruction model as the input for the reconstruction model in the next time step. As below we introduce two fusion methods: BDM-M (Merging) and BDM-B (Blending). Both are carried out under fusion-with-diffusion.

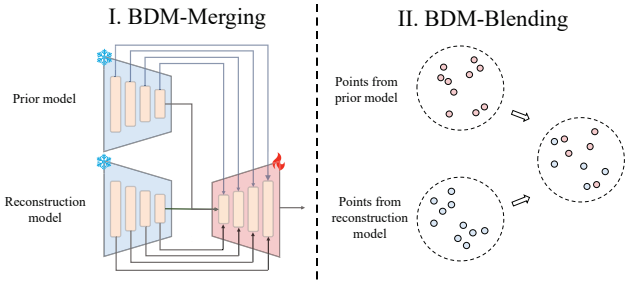

Figure 4. Illustration of our proposed fusion methods: BDM-M and BDM-B. The left part is the BDM-M, while the right side shows the BDM-B.

3.4.1 BDM-M (Merging)

In Merging, we propose a learnable paradigm for incorporating knowledge from the prior model into the reconstruction model. We implement BDM-M by using two diffusion models, PVD [84] and PC^2 [51]. Both the diffusion models use PVCNN as the backbone for predicting noise. Our Merging method focuses on the decoder part of PVCNN. To preserve the original knowledge in the reconstruction model, we freeze the encoder and finetune the decoder of PC^2. Specifically, following [82], we retain the original encoders of both models while feeding the multi-scale features of PVD's encoder directly into PC^2's decoder. Each layer of the encoder enhanced by this integration is facilitated by a zero-initialized convolution layer. This setup enables a seamless and implicit merging of the knowledge from the prior model and the reconstruction model.

3.4.2 BDM-B (Blending)

Beyond the implicit incorporation of prior knowledge, we also introduce an explicit training-free fusion method on point clouds, termed as BDM-B. It explicitly combines two groups of point clouds, $\mathbf{y}^t = \{\mathbf{z}_i^t\}_{i=1}^N$ and $\mathbf{y}_\gamma^t = \{\mathbf{z}_{\gamma,i}^t\}_{i=1}^N$, each consisting of N points. These point clouds are noise-reduced versions derived from the same origin, generated by generation and reconstruction models separately. The blending operation employs a probabilistic function Ψ, which assigns a selection probability to each pair of corresponding points $\mathbf{z}_j^t, \mathbf{z}_{\gamma,j}^t$, where $\mathbf{z}_i^t$ denotes the i-th point in a point cloud $\mathbf{y}$. The blending equation is defined as follows:

$$\mathbf{y}_\pi^t = \Psi(\mathbf{y}^t, \mathbf{y}_\gamma^t) \tag{6}$$

Additionally, assuming points are i.i.d., the whole point clouds can be formulated as

$$p\left(\mathbf{y}^t \mid \mathbf{y}^{t+1}\right) = \prod_{i=1}^{N} p\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{t+1}\right) \tag{7}$$

and

$$p_\gamma\left(\mathbf{y}^t \mid \mathbf{y}^{t+1}, \mathbf{x}\right) = \prod_{i=1}^{N} p_\gamma\left(\mathbf{z}_i^t \mid \mathbf{z}_i^{t+1}, \mathbf{x}\right) \tag{8}$$

Points are selected based on Ψ; for instance, we might choose 50% of the points from the prior and 50% from the reconstruction model. Statistically, this approach is akin to **blending** two point clouds, resulting in a mixed distribution of points from both sources. The blending method also supports our Bayesian Diffusion Model theory, the details of which will be given in the Appendix.

4. Experiment

Dataset. To demonstrate the efficacy of our Bayesian Diffusion Model, we conducted our experiments on two datasets: the synthetic dataset ShapeNet [12] and the real-world dataset Pix3D [63]. ShapeNet [9], a collection of 3D CAD models, encompasses 3,315 categories from the WordNet database. Following prior work [15, 51, 78], we utilized a subset of three ShapeNet categories – {*chair, airplane, car*} – from 3D-R2N2 [12], including image renderings, camera matrices, and train-test splits. Pix3D comprises diverse real-world image-shape pairs with meticulously annotated 2D-3D alignments. For a balanced comparison with CCD-3DR [15] and PC^2 [51], we reproduced these two works and adhered to CCD-3DR's benchmark on three categories: {*chair, table, sofa*}, allocating 80% of samples for training and the remaining 20% for testing. Further details about extended object categories are discussed in the Appendix.

Implementation Details. For both the ShapeNet-R2N2 and Pix3D datasets, we sample 4,096 points per 3D object and set the rendering resolution to 224×224. Notably, for Pix3D, images are cropped using their bounding boxes, necessitating adjustments to the camera matrices to accommodate the non-object-centric nature and varying sizes of the images. For the training of the generative diffusion model, we employ the PVD [84] architecture, adhering to their training methodologies. For the training of the reconstruction diffusion model, we select PC^2 and CCD-3DR as two baselines and follow the recipe of CCD-3DR. The inference step is set as 1,000 for both the generative model and the reconstruction model. In our BDM inference framework, Bayesian integration is strategically applied at specific intervals during the denoising process. This integration occurs every 32 steps, both in the early stage and in the late stage of denoising. The fusion process initiated by this integration extends throughout 16 steps, ensuring a balanced and effective incorporation of Bayesian principles throughout the denoising procedure. Training of generative models was conducted on 4 NVIDIA A5000 GPUs, while we trained reconstruction models utilizing a single NVIDIA A5000 GPU. More details are available in the Appendix.

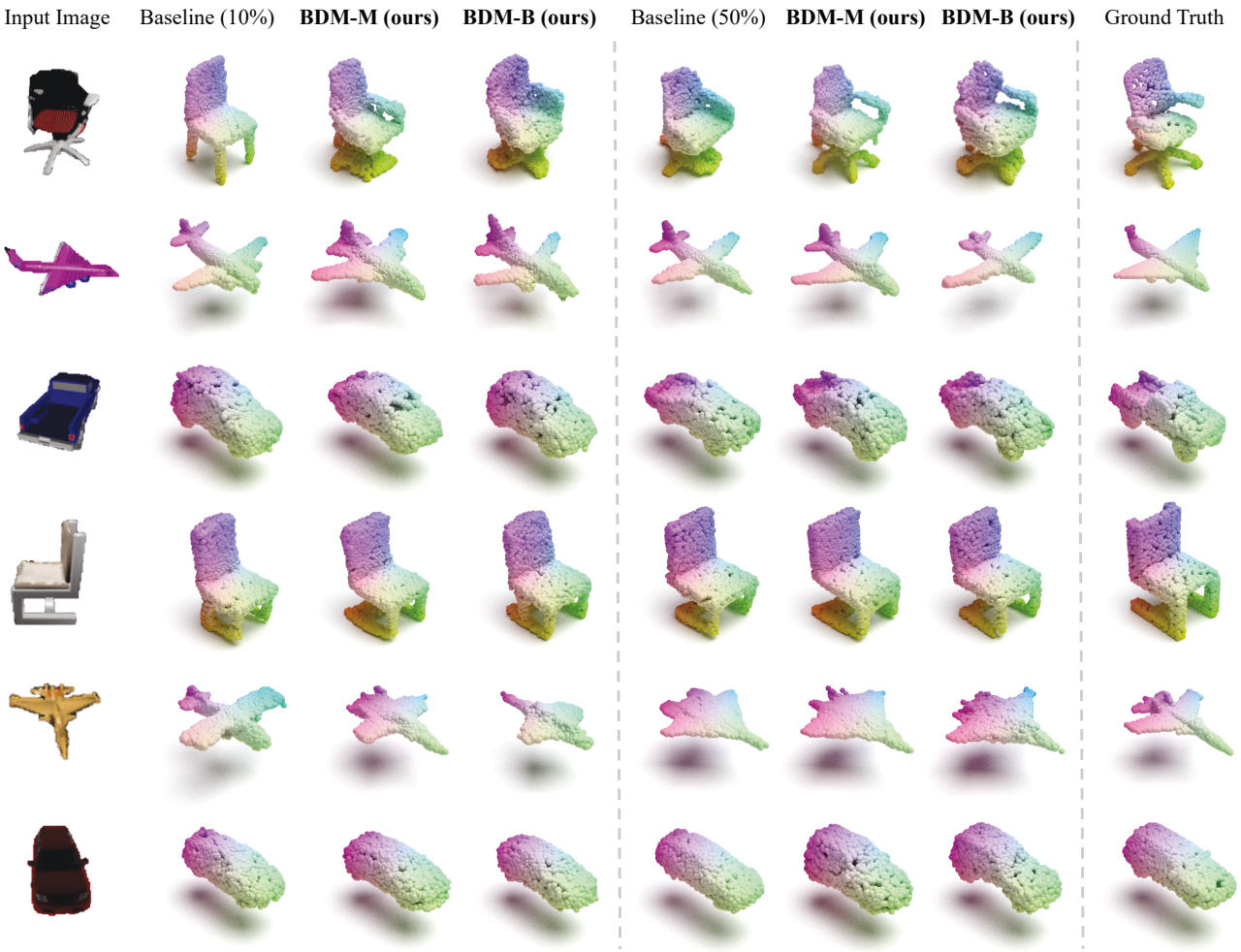

Figure 5. Qualitative comparisons on the synthetic ShapeNet-R2N2 dataset. We use PC^2 [51] and CCD-3DR [15] as baselines of 3D shape reconstruction. Rows 1-3 show the visualization of PC^2 while rows 4-6 display the result of CCD-3DR. We show our BDM's results under 10% data in column 2-4 and the results under 50% in column 5-7. Column 8 gives the corresponding ground truth.

4.1. Quantitative Results

We evaluate the performance of reconstruction with two widely recognized metrics: Chamfer Distance (CD) and F-Score@0.01 (F1). Chamfer Distance measures the disparity between two point sets by calculating the shortest distance from every point in one set to the closest point in the other set. To address the issue of CD's susceptibility to outliers, we additionally present F-Score at a threshold of 0.01. In this metric, a reconstructed point is deemed accurately predicted if its nearest distance to the points in the ground truth point cloud is within the specified threshold, which presents a measure of precision in the reconstruction process.

ShapeNet-R2N2. Tab. 1 presents our BDM's performance across various training data scales on ShapeNet-R2N2. The results indicate improvement in both CD and F1 across all three categories. Notably, the improvement becomes more pronounced when training data scale decreases, highlighting the effectiveness of the prior facing data scarcity.

Pix3D. Following CCD-3DR, we also evaluate on the Pix3D dataset in Tab. 2. It can be seen that our method effectively improves the performance and achieves state-of-the-art. Notably, we train the reconstruction model trained on Pix3D, while the prior model is trained on the same category of the separate, larger dataset: ShapeNet-R2N2 ($\sim$10 times size of Pix3D). In this way, we avoid the possible leaking and memorization problem, i.e., the generative model naively retrieve the nearest memorised shape learned from the reconstruction model.

4.2. Qualitative Results

In addition to the quantitative study, we also show the qualitative results to show the superiority of our BDM on both ShapeNet and Pix3D. For the ShapeNet-R2N2 dataset,

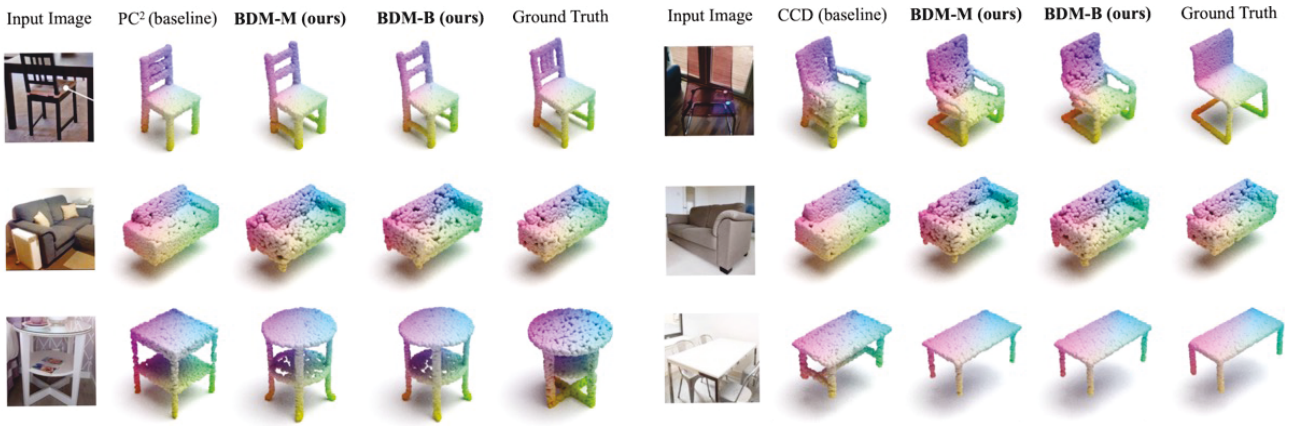

Figure 6. Qualitative comparisons on the real-world Pix3D dataset. We examine three distinct categories, each represented in a separate row. Columns 3,4 and 8,9 feature our BDM, implemented based on PC^2 and CCD-3DR respectively, for a comparative analysis.

Method	Chair						Airplane						Car					
	10%		50%		100%		10%		50%		100%		10%		50%		100%	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
Base: PC^2 [51]	97.25	0.393	73.58	0.437	65.57	0.464	88.00	0.605	76.39	0.628	65.97	0.655	64.99	0.524	62.59	0.542	64.36	0.547
BDM-M (ours)	94.94	0.395	71.56	0.446	64.48	0.468	87.75	0.604	73.19	0.629	65.16	0.653	63.53	0.524	**60.71**	0.549	**64.16**	0.554
BDM-B (ours)	**94.67**	**0.410**	**69.99**	**0.463**	**64.21**	**0.485**	**83.62**	**0.612**	**68.66**	**0.641**	**59.04**	**0.660**	**60.48**	**0.539**	62.58	**0.554**	65.85	**0.559**
Base: CCD-3DR [15]	89.79	0.418	63.13	0.474	58.47	0.498	81.29	0.612	72.46	0.635	62.77	0.651	63.13	0.531	62.25	0.550	**61.88**	0.562
BDM-M (ours)	81.47	0.425	62.07	0.477	57.31	0.493	81.54	0.608	69.31	0.632	61.87	0.652	61.92	0.531	61.24	0.555	63.66	0.561
BDM-B (ours)	**79.26**	**0.441**	**60.07**	**0.497**	**56.78**	**0.510**	**77.34**	**0.621**	**66.83**	**0.644**	**56.96**	**0.660**	**59.05**	**0.546**	**60.59**	**0.560**	66.90	**0.569**

Table 1. Performance on *Chair, Airplane* and *Car* of ShapeNet-R2N2. We evaluate our BDM, comparing with two baselines: PC^2 and CCD-3DR. These experiments span three different scales of training data (10% / 50% / 100%) for the reconstruction diffusion model, demonstrating the efficacy of our BDM method.

Method	Chair		Sofa		Table	
	CD↓	F1↑	CD↓	F1↑	CD↓	F1↑
Base: PC^2 [51]	115.94	0.443	47.17	0.445	202.77	0.397
BDM-M (ours)	113.40	0.449	**44.50**	0.451	202.08	0.413
BDM-B (ours)	**110.60**	**0.455**	45.05	**0.455**	**186.46**	**0.429**
Base: CCD-3DR [15]	111.42	0.456	44.91	0.450	196.28	0.418
BDM-M (ours)	**110.86**	0.464	44.86	0.452	**193.56**	**0.424**
BDM-B (ours)	111.12	**0.466**	**44.51**	**0.456**	194.91	0.423

Table 2. Performance on *Chair, Sofa* and *Table* of Pix3D. We evaluate our BDM on the two baselines: PC^2 and CCD-3DR.

Fig. 5 demonstrates our BDM's capacity in synthetic 3D object reconstruction. In Fig. 6, it can be seen clearly that our method surpasses baselines with respect to the reconstruction quality on the Pix3D dataset. Notably, our BDM effectively restores the missing spindles of the chair in the first image and eliminates hallucinations in the last image.

4.3. Efficiency and Fairness Analysis

As shown in Tab. 3, we present BDM's parameters, runtime and GPU memory when doing inference with batch size of 1. While parameters increase due to the incorporation of prior model $\mathcal{P}$, memory usage and runtime of BDM only increase slightly. Moreover, BDM leverages off-the-shelf pre-trained weights for prior model $\mathcal{P}$ which is **frozen**, which doesn't further incur additional training cost.

	PC^2/CCD-3DR	BDM-B	BDM-M
#Parameters (M)	47.41	73.78	74.82
Runtime (s)	46.89	48.84	49.24
GPU memory (GB)	1.73	1.93	2.01

Table 3. Model parameters and inference-time efficiency.

4.4. Ablation Study

To validate the effectiveness and rationality of our BDM, we conduct several ablation studies to explore the impact of the timing, duration and intensity to absorb priors. Unless otherwise specified, we set BDM-B comprising PVD trained on 100% data and CCD-3DR trained on 10% data from ShapeNet-*chair* as our baseline.

Prior Integration Timing. This subsection evaluates the effectiveness of integrating prior knowledge at various stages of the denoising process. Tab. 4 reveals that integrating priors in the late stage alone yields significant improvement on model performance, reducing CD to 80.22 and increasing F1 to 0.436. Furthermore, combining early and late-stage integration further enhances results, achieving the lowest CD of 79.26 and the highest F1 of 0.441. This

contrasts with the middle stage integration, which even degrades the performance of the baseline on F1.

Early	Middle	Late	Chamfer Distance ↓	F-Score@0.01 ↑
			89.79	0.418
✓			82.34	0.421
	✓		87.61	0.395
		✓	80.22	0.436
✓		✓	**79.26**	**0.441**
✓	✓	✓	81.92	0.416

Table 4. Ablation on the timing of prior integration. This table presents the impact of applying prior integration during the early, middle, and late stages of the denoising process on CD and F1.

Prior Integration Duration. As shown in Tab. 5, we investigate how varying the duration of prior integration affects the denoising process. First, the performance will greatly improve once the prior duration is greater than 1, thereby strongly validating the effectiveness of our BDM. Notably, the prior integration duration of 16 steps demonstrates the most substantial improvement. This is a remarkable improvement over the baseline (0 step). The diminishing returns are observed with the duration of 32 steps, suggesting that a moderate duration of prior integration optimally balances denoising effectiveness and prior guidance.

Prior Duration	0 step (baseline)	1 step	2 step	4 step	8 step	16 step	32 step
Chamfer Distance ↓	89.79	80.89	81.02	80.65	79.72	**79.26**	79.94
F-Score@0.01 ↑	0.418	0.427	0.430	0.429	0.432	**0.441**	0.438

Table 5. Ablation on the duration of prior integration. This table presents how different durations of prior integration affect CD and F1, ranging from 0 to 32 steps.

Prior Integration Ratio. In this part, we present an ablation on the impact of the prior integration ratio on the BDM-B process. As can be seen in Tab. 6, interestingly, a gradual increase in the prior take-in ratio yields varying results. At 25%, there is a minor detriment to performance. However, at 50%, we observe a significant improvement, marking the optimal balance in prior integration. On the contrary, further increasing the ratio to 75% and 100% leads to a drastic decline in performance. These results suggest that while a moderate level of prior integration enhances the model's performance, excessive integration can be detrimental.

Prior Ratio	0% (baseline)	25%	50%	75%	100%
Chamfer Distance ↓	89.79	88.22	**79.26**	142.69	256.13
F-Score@0.01 ↑	0.418	0.382	**0.441**	0.307	0.242

Table 6. Ablation on the ratio of prior integration. This table compares the effects of different prior integration ratios on CD and F1.

4.5. BDM vs CFG

Considering that Classifier-Free Guidance (CFG) [30] also shows the relation in the inference stage between the gradient $\varepsilon(\mathbf{y}^t)$ from the unconditional diffusion model and the gradient $\varepsilon_\gamma(\mathbf{y}^t, \mathbf{x})$ from the conditional diffusion model, we conduct an ablation study to compare our method with CFG. As shown in Tab. 7, both our methods surpass CFG.

	Chamfer Distance ↓	F-Score@0.01 ↑
Baseline	58.47	0.498
CFG [30]	59.08	0.495
BDM	**56.78**	**0.510**

Table 7. We compare our BDM with CFG. We train these models and test these two methods on 100% data of chair from ShapeNet.

4.6. Human Evaluation

To better evaluate the reconstruction quality, we also conducted human evaluation. We randomly selected 20 comparison groups from the Chair, Airplane, and Car classes in the ShapeNet dataset, totaling 60 groups. In each group, we present outputs generated by CCD-3DR, BDM-B, and BDM-M. Sixteen evaluators then ranked each group on a scale of 1 to 3, and the average scores are shown in Tab. 8. The results show that our two methods, BDM-M and BDM-B, still outperforms CCD-3DR, which is aligned with the quantitative result presented in the main paper. More details will be discussed in the Appendix.

Human Evaluation	CCD	BDM-B	BDM-M
Chair	1.48	2.15	2.32
Airplane	1.74	1.86	2.40
Car	1.77	1.88	2.35
Average	1.67	1.96	2.35

Table 8. Human Evaluation over 3 categories for CCD, BDM-B, and BDM-M, with 3 being the best and 1 being the worst.

5. Conclusion and Limitations

In this paper, we present Bayesian Diffusion Model (BDM), a novel diffusion-based inference method for posterior estimation. BDM overcomes the limitations in the traditional MCMC-based Bayesian inference that requires having the explicit distributions in performing stochastic gradient Langevin dynamics by tightly coupling the bottom-up and top-down diffusion processes using learned gradient computation networks. We show a plug-and-play version of the BDM (BDM-B) and a learned fusion version (BDM-M). BDM is particularly effective for applications where paired data-labels such as image-object point clouds, are scares, while standalone labels, like object point clouds, are abundant. It demonstrates the state-of-the-art results for single image 3D shape reconstruction. BDM points to a promising direction to perform the general inference in computer vision and machine learning beyond the 3D shape reconstruction application shown here.

Limitations. BDM requires both the prior and data-driven processes to be diffusion processes. Also, BDM-B takes advantage of the explicit representation of point clouds, which might not be broadly adopted on implicit representations.

Acknowledgement. This work is supported by NSF Award IIS-2127544. We are grateful for the constructive feedback by Zirui Wang and Zheng Ding.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, 2018. 2

[2] Titas Anciukevičius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J Mitra, and Paul Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *CVPR*, 2023. 2

[3] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003. 1, 3

[4] Joseph J Atick, Paul A Griffin, and A Norman Redlich. Statistical approach to shape from shading: Reconstruction of three-dimensional face surfaces from single two-dimensional images. *Neural Computation*, 8(6):1321–1340, 1996. 2

[5] José M Bernardo and Adrian FM Smith. *Bayesian Theory*. 2009. 1, 2, 3

[6] Andrew Blake and Michael Isard. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. 2012. 1, 2

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 3(Jan):993–1022, 2003. 1

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 2

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 5

[10] German KM Cheung, Simon Baker, and Takeo Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *CVPR*, 2003. 2, 3

[11] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*, 2020. 2

[12] 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 *ECCV*, 2016. 2, 5

[13] Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61 (1):38–59, 1995. 1, 2

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[15] Yan Di, Chenyangguang Zhang, Pengyuan Wang, Guangyao Zhai, Ruida Zhang, Fabian Manhardt, Benjamin Busam, Xiangyang Ji, and Federico Tombari. Ccd-3dr: Consistent conditioning in diffusion for single-image 3d reconstruction. *arXiv preprint arXiv:2308.07837*, 2023. 2, 5, 6, 7

[16] 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. In *ICLR*, 2021. 1, 2

[17] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 2

[18] Li Fei-Fei and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 1, 2

[19] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106: 59–70, 2007. 1

[20] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61:55–79, 2005. 1, 2

[21] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 1, 2

[22] Vincent Fortuin. Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3):563–591, 2022. 1

[23] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *ICCV*, 2021. 2

[24] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *3DV*, 2017. 2

[25] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*, 2016. 2

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2

[27] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007. 1

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[29] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *ICCV*, 2019. 2

[30] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 3, 8

[31] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3

[32] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970. 2

[33] Long Jin, Justin Lazarow, and Zhuowen Tu. Introspective classification with convolutional nets. *NeurIPS*, 2017. 2

[34] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2): 181–214, 1994. 1

[35] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018. 2

[36] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *CVPR*, 2015. 2

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[38] David C Knill and Whitman Richards. *Perception as Bayesian inference*. 1996. 1, 2

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1, 2

[40] John D Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 1, 3

[41] Stan Z Li. *Markov random field modeling in image analysis*. 2009. 1

[42] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 2

[43] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010. 1, 2

[44] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Zexiang Xu, Hao Su, et al. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In *NeurIPS*, 2023. 2

[45] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *ICCV*, 2023. 2

[46] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *NeurIPS*, 2019. 2

[47] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 2

[48] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *CVPR*, 2021. 2

[49] Priyanka Mandikal and Venkatesh Babu Radhakrishnan. Dense 3d point cloud reconstruction using a deep pyramid network. In *WACV*, 2019. 2

[50] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. 2010. 1

[51] Luke Melas-Kyriazi, Christian Rupprecht, and Andrea Vedaldi. Pc2: Projection-conditioned point cloud diffusion for single-image 3d reconstruction. In *CVPR*, 2023. 2, 3, 5, 6, 7

[52] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2

[53] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. 2022. 2

[54] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729, 2008. 1

[55] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[56] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 2

[57] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 2

[58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1, 2

[59] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2

[60] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *ICCV*, 2019. 2

[61] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015. 2, 3

[62] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020. 2, 3

[63] 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 *CVPR*, 2018. 2, 5

[64] Zhuowen Tu. Learning generative models via discriminative approaches. In *CVPR*, 2007. 2

[65] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *TPAMI*, 24(5):657–673, 2002. 1

[66] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *IJCV*, 63:113–140, 2005. 1, 2

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[68] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 1

[69] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 2

[70] Markus Weber, Max Welling, and Pietro Perona. Unsupervised learning of models for recognition. *Lecture Notes in Computer Science*, 2000. 1, 2

[71] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *ICML*, 2011. 1, 3

[72] Andrew P Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 17(1-3):17–45, 1981. 2

[73] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *TPAMI*, 31(12):2115–2128, 2009. 1, 2

[74] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, 2016. 2

[75] Yuefan Wu, Zeyuan Chen, Shaowei Liu, Zhongzheng Ren, and Shenlong Wang. CASA: Category-agnostic skeletal animal reconstruction. In *NeurIPS*, 2022. 2

[76] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, 2015. 2

[77] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *ICCV*, 2019. 2

[78] Haozhe Xie, Hongxun Yao, Shengping Zhang, Shangchen Zhou, and Wenxiu Sun. Pix2vox++: Multi-scale context-aware 3d object reconstruction from single and multiple images. *IJCV*, 128(12):2919–2935, 2020. 5

[79] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *ICCV*, 2019. 2

[80] Alan Yuille and Daniel Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–308, 2006. 1

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

[82] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 5

[83] Xiang Zhang, Zeyuan Chen, Fangyin Wei, and Zhuowen Tu. Uni-3d: A universal model for panoptic 3d scene reconstruction. In *ICCV*, 2023. 2

[84] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *ICCV*, 2021. 2, 3, 5