Learning to Iteratively Improve 3D Representation with 2D Generative Models

Anonymous Author(s)

Abstract

Reconstructing three-dimensional (3D) representations from sparse image data is a core task that requires learning to sample plausible 3D models that correspond to 2D conditioning images. Despite numerous proposed frameworks, achieving photorealistic sparse-view 3D reconstructions remains an unresolved challenge, with current methods often producing blurry results on small object-centric scenes that fall short of the fidelity achieved by dense-view 3D reconstruction and 2D generative models. This paper aims to rethink the use of image generative models for 3D reconstruction and introduces a novel framework based on iterative refinement. Our approach infers the 3D representation by optimizing it to match images sampled by a 2D generative model, itself conditioned on the current progress of the 3D optimization. To learn this conditional generative model, we design a new training strategy that performs 3D reconstruction using various numbers of views and captures the progress at each optimization timestep. This allows the model to explicitly learn to sample images that are consistent with the current stage of 3D reconstruction, supporting sampling of thousands of consistent images during reconstruction. Experiments on a challenging real-world dataset demonstrate competitive performance in single-view 3D reconstruction, performing on par with state-of-the-art 3D reconstruction methods based on 2D generative model outputs and dense mulitview images.

1 Introduction

Reconstructing a three-dimensional (3D) representation of the physical world from sparse signals, such a two-dimensional (2D) image, is a fundamental task in the fields of computer vision, graphics, and artificial intelligence. Such representations are crucial for applications in augmented and virtual reality (AR/VR) as they allow rendering from novel viewpoints, and in navigation, robotics and AI, as they support reasoning about object extents and the world around us. However, despite the plethora of frameworks proposed in the last decades, the reconstruction of 3D scenes from one or few images remains an unresolved problem.

The challenge lies in the inherent ambiguity of the task: multiple 3D scenes can correspond to a single 2D image, and even more possibilities exist for the unbounded space outside the region seen in the image. Consequently, photogrammetry methods, including recent methods based on neural networks [48], which reconstruct a 3D scene using large dataset of images, fail when only few images are available, as they cannot sample plausible content in regions that are unobserved in the input images. More formally, the reconstruction task is probabilistic and generative in nature—its solution is a plausible 3D sample out of many possible ones, requiring learning the model capable of sampling from a posterior probability distribution conditioned on one or more input images.

Inspired by the progress of generative models of images and videos, the last decade of 3D research has investigated using neural networks to learn a prior about how 3D scenes should look. However, unlike in 2D datasets that are easy to collect from widely available images on the internet, large datasets of unbounded 3D scenes are infeasible to create. Therefore, the research community is on a quest to find algorithms that learn a prior over the 3D world from only multi-view image datasets, such as ones captured from a consumer camera [61]. Two dominant streams of research have emerged; each, however, produces blurry reconstructions, significantly lagging behind techniques that utilize

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

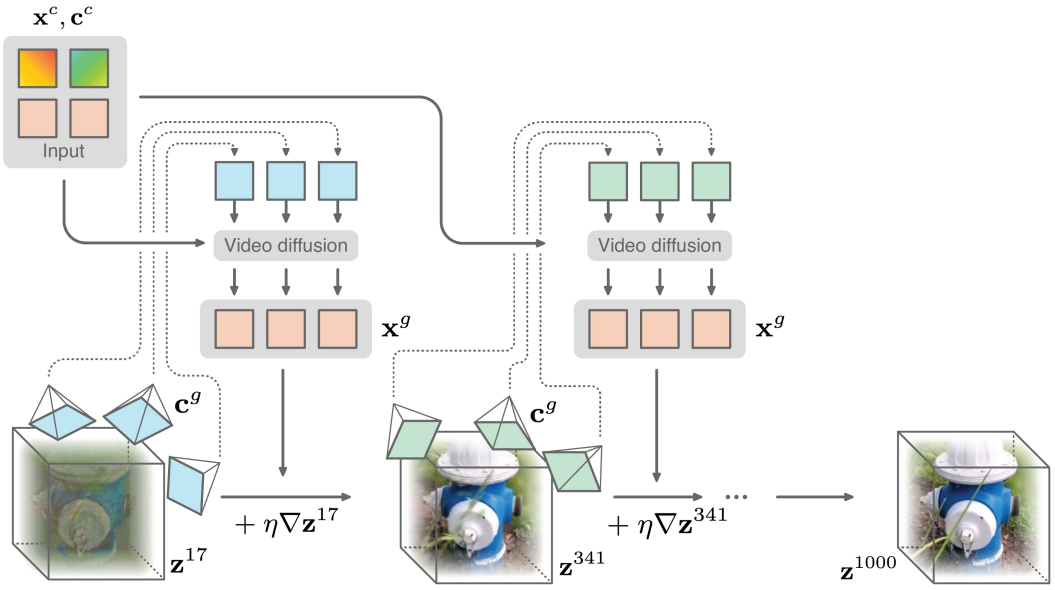

Figure 1: Proposed framework for 3D reconstruction. Given input views and poses, $\mathbf{x}^c, \mathbf{c}^c$, our framework uses generative image model to generate images $\mathbf{x}^g$ at novel views $\mathbf{c}^g$, conditioned on 3D representation $\mathbf{z}^t$ at each optimization step t. An optimization step is taken at each iteration, where the gradient $\nabla\mathbf{z}^t$ is computed by rendering representation $\mathbf{z}^t$ to generated image viewpoints $\mathbf{c}^g$. In contrast to prior works where images are generated in one-shot manner, our approach allows sampling thousands of images over time, each increasingly consistent with each other.

dense sets of views, such as Gaussian Splatting, and those for photorealistic 2D image generation, like image diffusion models [6]. Some works have proposed 3D-aware generative models [3, 9, 38] which learn to model 2D images by rendering a 3D representation. These methods hand-engineer differentiable rendering into the probabilistic model, enforcing the model to learn a prior over this 3D representation. However, hand-engineered representations and rendering operators cannot perfectly capture real-world scenes and limits the capacity of the generative model. Some works learn black-box generative models to generate images from novel viewpoints [43, 18], and afterwards use many-view photogrammetry to reconstruct 3D from generated dataset of images. However, for a reconstruction of an unbounded 3D scene, photogrammetry methods require hundreds of images, which current models cannot generate [18]. Moreover, such methods generate images in one shot, often resulting in inconsistencies, which lead to blurry results from the 3D reconstruction stage.

In this work, we rethink how to best use image generative models for the task of 3D reconstruction, and introduce a novel framework for sampling a 3D reconstruction given one or few images. Similarly to most recent works, we use general many-view 3D reconstruction methods operating on generated images. However, instead of sampling a dataset of images in one shot, our probabilistic model samples images iteratively, over the course of optimization, each consistent with the current state of the 3D representation. Unlike 3D-aware generative models, our method supports unbounded scenes of unlimited resolution and can integrate any number of conditioning images. Unlike previous 2D generative models, our method can generate thousands of images consistent with each other and with the 3D representation, supporting reconstruction of large scenes. This framework effectively decouples representation from inference, making it scalable and general-purpose [73], allowing it to be used as plug-and-play component on any existing 3D reconstruction pipeline, such as Gaussian Splatting [34] or NeRFs [48]. In our experiments, we instantiate this framework using Gaussian splats as the representation and video diffusion as the generative model; we demonstrate state-of-the-art performance in sparse view 3D reconstruction, surpassing competing frameworks of latent variable generative models and 3D reconstruction from 2D generative model outputs.

2 Prior Methods

In this section, we analyse various frameworks that have been proposed to reconstruct 3D scene representations from images. We first (Sec. 2.1) summarise various methods for representing and

rendering the 3D world, which can be inverted to reconstruct from a dense set of images. We then discuss (Sec. 2.2) generative models which can sample 3D scenes given sparse images and then review generative 3D models that learn to sample 3D representations whilst learning from 2D images.

2.1 3D Representation and Rendering

Computer graphics has developed methods for representing the physical 3D world and simulating the image formation process via rendering [32], enabling generation of realistic images. The core idea of inverse graphics is that given a dataset D of images and their poses $\{(\mathbf{x}^i, \mathbf{c}^i) \mid i = 0, \ldots, N\}$, the process of rendering can be "inverted" to infer a 3D representation $\mathbf{z}$ that generated the images. This is achieved by optimization that minimizes reconstruction a loss where gradients with respect to the 3D representation $\nabla\mathbf{z}$ are calculated using a differentiable rendering function:

```
D ← {(x^i, c^i) | i = 0, ..., N}
def reconstruct(D, T, render(), z^0):
    for t in range(T):
        x^g, c^g ← random.choice(D)
        ∇z^t ← render(z^t, c^g).loss(x^g).grad
        z^{t+1} ← z^t + η∇z^t
    return z^T
```

Under certain assumptions, such as a large number of input images N, such inference process results in a good 3D representation that can then be rendered to novel views. Over the years many representations and rendering algorithms have been proposed, the most popular being surface representations (such as distance fields and polygon meshes) that can be rendered to images by rasterization or path tracing. Since meshes are difficult to optimize using gradient descent due to non-local gradients, neural radiance fields (NeRFs) [48] have been introduced, which represent a volume with a neural network. More recent works, such as iNGP [51] and 3D Gaussian Splatting [34, 39], have focused on increasing the speed of training and rendering to real-time. Another direction is aimed to acquire physically meaningful representations [55]. However, a core limiting assumption is access to large amount of training images (e.g. capturing every side of the object), typically requiring $N > 100$ for a single room and $N > 1000$ for multi-room scenes. Consequently, when such an amount of images is not feasible to acquire in practice, these methods produce floating artifacts and empty volumes in under-sampled regions of 3D space. Some methods aim to fix reconstruction errors using regularizers on depth, normal, or colors, or by discriminators and image generative models [54, 63, 44, 31]. However, these approaches already assume access to a fully reconstructed 3D model.

2.2 Generative Models

Generative models learn to sample from the complex distribution of their training data. Various families have been proposed including Generative Adversarial Networks (GAN) [19], Variational Autoencoders (VAE) [62, 36], autoregressive models [81, 80, 60] and Independent Component Analysis (ICA) [30]. Recent success in high-dimensional data, such as images [64], videos and sound, have been achieved by score-based generative models [69, 71, 70], particularly denoising diffusion probabilistic models (DDPM) [26, 72]. These learn to estimate the gradient $\nabla\mathbf{z}^t$ of the log probability $p(\mathbf{z}, t)$ (termed "score") with respect to the data at a noise scale t. Inspired by their success in other modalities, DDPMs have been adopted to sample 3D representations by learning from datasets of ground-truth 3D representations [11, 14], such as pointclouds [46, 79], Neural Fields [5, 50, 29, 40, 12, 84, 35, 68, 21, 33, 20] or 3D Gaussians [89, 49]. At test-time, these methods support conditioning on input views $\mathbf{x}^c$ and poses $\mathbf{c}^c$, and sampling a plausible 3D reconstruction $\mathbf{z}$:

```
def reconstruct(x^c, c^c, T, P_θ, σ):
    z^0 ← random()
    for t in range(T):
        ∇z^t ← P_θ(z^{t+1} | z^t, x^c, c^c).sample()
        z^{t+1} ← α^t z^t + η^t ∇z^t
    return z^T
```

We similarly aim to perform flow matching between randomly sampled 3D representations and empirical distribution of 3D representations. However, unlike 2D images, large datasets of highly-realistic and large-scale 3D scenes are challenging or even infeasible to create. Therefore, we propose a method that learns to sample 3D scenes whilst learning from widely available 2D image datasets.

Structure-in. Some methods aim to learn to sample latent 3D representations whilst learning to generate 2D images. These methods typically have a 3D representation inside their architecture, hence often denoted as “3D-aware" or “structure-in", as the 3D representation and rendering are hand-engineered inside the network. These models define a likelihood over images by sampling a latent variable corresponding to a 3D representation and then rendering it to an image. For example, 3D-aware diffusion [3] learns to denoise image via underlying 3D representation, 3D-aware VAEs learn a latent variable model where latent variable is a 3D representation [38, 23, 25, 1, 24] and 3D-aware GANs learn a generator that generates images by first generating a 3D representation [53, 66, 9, 15, 52, 16]. This framework has been extended to in-the-wild datasets [2, 87, 45, 28, 74, 78, 27, 8]. However, the core limitation of these models is that 3D representation and rendering have to be hand-engineered into the model. This restricts the flexibility and capacity of the model, as the representations are of limited flexibility and the rendering operation is only approximate. For example, current state-of-the-art generative methods use representations having a limited number of parameters, such as voxel grids, triplanes, or image-supported features. Furthermore, their renderers only consider the final bounce of light from one surface to the camera, e.g. without modelling reflections. Some approaches replace hand-engineereed rendering by a learnt “neural" renderer [17, 10], however at the cost of losing the 3D representation that is needed in many applications.

Structure-out. Instead of hand-engineering the 3D representation and rendering inside the generative model, some approaches try to extract 3D structure from 2D images generated by black-box generative models. The most straightforward approach is to generate a dataset of images and poses $D = \{(\mathbf{x}^i, \mathbf{c}^i) \mid i = 0, \dots, N\}$ using a generative video model and then run an optimization-based 3D reconstruction method as described in Sec 2.1. This has the benefit that advancements in graphics can be utilised out-of-the box, e.g. by using unconstrained and flexible 3D representations with reflection-aware rendering, which avoids hand-engineering generative image models, and allows flexible and powerful architectures trained on large amounts of 2D datasets. Most recent works [22, 82, 18, 86, 47, 90, 43, 77, 45, 37, 42, 75, 6] use image diffusion models fine-tuned with camera-pose and then generate a dataset of 2D images from which 3D is reconstructed. However, the classic 3D reconstruction methods assume access to large amount of 3D consistent images N. In contrast, current approaches generate images that are slightly inconsistent, both due to limited performance of generative models and due to camera pose conditioning being incorrect. Consequently, this leads the 3D reconstruction method to “average out" these inconsistencies, resulting in blurry regions. Another problem is that generating hundreds or thousands of images is not possible with current multi-view generative models. Instead, current approaches generate small sets of images conditionally independently from each other, which results in inconsistent 3D scenes even assuming access to a perfect generative model. A concurrent work [18] generates images in sets of 8, first generating a set of anchor frames and then autoregressively generating the rest; it relies on ad-hoc techniques, such as using LPIPS loss [86, 18] to be invariant to inconsistent generated images. Consequently, current methods are limited to small and bounded object-centric scenes where small number of images suffice. In this work, we use generative image models to generate thousands of consistent images by explicitly training the model to output images that are consistent with previous generations.

3 Method

Our method tackles the problem of reconstructing a 3D representation from a small number of input images. The proposed framework modifies only one line in the classical 3D reconstruction pipeline (Sec. 2.1) – instead of using dataset of images and poses, our framework samples images $\mathbf{x}^g$ iteratively throughout the optimization process. At each optimization step t, images $\mathbf{x}^g$ are sampled from a learnt probabilistic generative model P_θ conditioned on the current stage of reconstruction $\mathbf{z}^t$ and input (conditioning) images and poses $\mathbf{x}^c$, $\mathbf{c}^c$. Then, a gradient with respect to the representation $\nabla\mathbf{z}^t$ is computed by rendering representation to images:

def reconstruct($\mathbf{x}^c$, $\mathbf{c}^c$, T, P_θ, render(), $\mathbf{z}^0$):
  for t **in range**(T):
    $\mathbf{c}^g \leftarrow P_\lambda(\mathbf{c}^g \mid \mathbf{c}^c)$.sample()
    $\mathbf{x}^g \leftarrow P_\theta(\mathbf{x}^g \mid \mathbf{z}^t, \mathbf{x}^c, \mathbf{c}^g)$.sample()
    $\nabla\mathbf{z}^t \leftarrow$ render$(\mathbf{z}^t, \mathbf{c}^g)$.loss$(\mathbf{x}^g)$.grad
    $\mathbf{z}^{t+1} \leftarrow \mathbf{z}^t + \eta\nabla\mathbf{z}^t$
  return $\mathbf{z}^T$

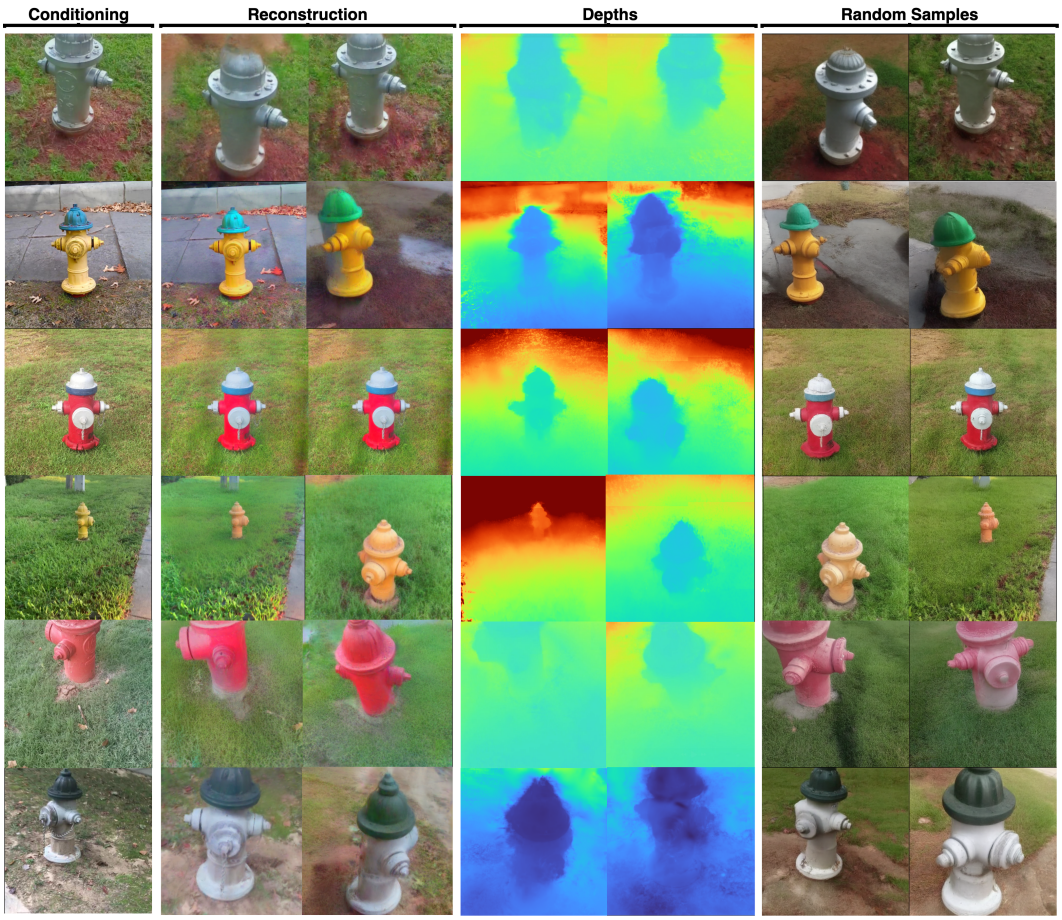

Figure 2: Qualitative results from our model on 3D reconstruction from a single image (first three rows) and six images (next three rows). The leftmost column shows the input (conditioning), followed by two novel views rendered from the reconstructed 3D representation and their corresponding depth maps. The final columns present samples from our generative model at an early stage t of optimization, illustrating its ability to generate diverse yet consistent images, each pushing the 3D representation closer to the true posterior sample.

Note that this framework separates 3D representation and rendering from the generative image model, allowing out-of-the-box use of advancements in graphics (e.g. fast optimization and real-time rendering of 3D Gaussian Splatting [34, 39]) and unconstrained architecture of generative model (e.g. diffusion or flow-based models). Importantly, at each step, the generative model is conditioned on the current stage of reconstruction $\mathbf{z}^t$, allowing to sample images that are consistent with the 3D scene and previous generations $t-1, t-2, ..., 0$. Note that we do *not* maximize likelihood, i.e., $P_\theta(\mathbf{x}^g \mid \mathbf{z}^t, \mathbf{x}^c, \mathbf{c}^g)$.likelihood(render$(\mathbf{z}^t, \mathbf{c}^g)$), as this would lead to mode-seeking optimization behavior, akin to score-distillation sampling [58, 85, 83, 76, 41], resulting in poor reconstruction quality when the conditioning datapoint does not reside near the modes of the distribution. In this section, we describe this framework by providing details on the generative model (Sec. 3.1), how it is conditioned on current stage of reconstruction (Sec. 3.3) and input (Sec. 3.4), model training (Sec. 3.5), and representation and rendering (Sec. 3.6).

3.1 Generative Model

At each optimization step t, we use a learnt probabilistic model to sample images that are consistent with both input conditioning and previously generated images. Specifically, the probabilistic model $P_\theta(\mathbf{x}^g \mid \mathbf{z}^t, \mathbf{x}^c, \mathbf{c}^c)$ samples images $\mathbf{x}^g$ at specific poses $\mathbf{c}^g$, conditioned on the current stage of reconstruction $\mathbf{z}^t$. The generation poses $\mathbf{c}^g$ are sampled in such way that minimizes the prediction entropy of the autoregressive chain (see Sec. 3.2). For the generative model, we adopt the framework of Latent Video Diffusion Models (LVDM) [64, 7, 57]. LVDMs employ Denoising Diffusion

Probabilistic Models [69, 26, 71] to generate latent variables, which are then decoded into multi-view images. For simplicity, latent representations are omitted in figures. During training, the model is trained to denoise target images $\mathbf{x}^g$, conditioned on target camera poses $\mathbf{c}^g$, conditioning input images $\mathbf{x}^c$, and the current stage of 3D reconstruction $\mathbf{z}^t$. The input to the denoising model consists of noisy video latents $\mathbf{x}^g$ with dimensions $[G, C, H, W]$, where G is the number of views, C is the number of channels, and H and W are the height and width of the image latents. The denoising diffusion model, parameterized by $\boldsymbol{\theta}$, is trained to predict the denoised latents from the noisy latents. During optimization, to compute the loss (as shown in line 7 of the pseudocode), we use images sampled from the trained posterior distribution, i.e. $\mathbf{x}^g \leftarrow P_\theta(\mathbf{x}^g \mid \mathbf{z}^t, \mathbf{x}^c, \mathbf{c}^g)$.sample().

3.2 Autoregressive Generation with Uncertainty-Guided Ordering

The camera poses where new images are being generated are sampled from $P_\lambda(\mathbf{c}^g \mid \mathbf{c}^c)$, which we have a control over. We observed that the choice of P_λ has a profound effect on the faithfulness of the generated images to the conditioning input as well as numerical reconstruction results. We observed that a naive choice of P_λ, e.g. sampling a starting point randomly around the object as in previous one-shot works, leads to query poses that are far from previously generated images or from input poses, where P_θ struggles to generate consistent images. In contrast, we found that P_θ can easily generate consistent images that are close to previous generations. Therefore, we discovered that the optimal ordering strategy is to query views that contain the least uncertainty, i.e. would exhibit only small variation in generations. Thus, we prefer views near to previous generations as opposed to views of the opposite side of the input image which contain a lot of uncertainty in unobserved regions. More formally, we aim for an ordering $x_1 \rightarrow x_2 \rightarrow x_3$ of view subsets x_i such that overall entropy is minimized:

$$H(\mathbf{x}) = H(x_1) + H(x_2 \mid x_1) + H(x_3 \mid x_1, x_2) \tag{1}$$

In practice, we first generate various videos (simulating similar camera motions as in the training data) around the input poses. These are then gradually expanded and the process repeats.

3.3 Conditioning on Current 3D Representation

We condition generative model P_θ on the current stage of 3D reconstruction $\mathbf{z}^t$ to allow learning to generate images that are consistent with previously generated images. We achieve this by rendering 3D representation (e.g. partially reconstructed 3D Gaussians) $\mathbf{z}^t$ to the same viewpoints as images to be generated $\mathbf{x}^g$ and encoding them with the latent diffusion's VAE to get another set of latents. As these latents are of the same dimensions, we concatenate them as extra channels with noisy latents and feed them together to the denoising U-Net. We found that such conditioning on current stage of reconstruction provides the model with a rich signal about previously generated images that were used for the reconstruction, as the 3D scene is seen from multiple viewpoints. Furthermore, conditioning our model on renderings provides rich information about camera extrinsics and intrinsics.

3.4 Conditioning on Input Images and Poses

We condition the generative model on input images and poses. Previous methods have utilized CLIP conditioning, which leverages features from a large pretrained model optimized for image-to-text matching [59]. While these features are semantically rich, they may lack detailed information about high-frequency scene elements, such as precise object shapes and textures. To address this, we additionally condition the model on DINOv2 features [56], which extract 16x16 spatial tokens and a global token. However, using all tokens directly is computationally expensive. Therefore, we pool the 256 spatial tokens into a single token and concatenate it with the global DINO token before combining them with CLIP embeddings. During training, we condition on a variable number of input images, allowing the denoising U-Net to cross-attend over these tokens.

The model must also understand the relationship between input views and the views it needs to generate. To achieve this, we add camera pose embeddings and sum them with the DINOv2 features. To avoid providing duplicate pose information to the generative model, the conditioning poses $\mathbf{c}^g$ are made relative to the first generated image—i.e., the images provided to the generative model are always assumed to start at an identity pose. We then perform positional embedding of camera poses and sum them with other tokens, enabling the U-Net to cross-attend to image tokens based on their poses.

Note that by retaining the classical 3D reconstruction, our framework naturally supports another pathway for conditioning on an arbitrary number of images, as we can pass them as additional images to be reconstructed. This is a capability that "structure-in" methods cannot easily achieve (Sec. 2.2).

3.5 Training for Iterative Reconstruction

We use 2D video datasets to train our probabilistic model. However, using 2D image datasets directly for training is not feasible as our model requires conditioning on the reconstruction $\mathbf{z}^t$ at optimization step t. Therefore, we construct a dataset by performing classical 3D reconstruction (e.g. Gaussian Splatting or NeRF as in Sec. 2.1) from varying numbers of input images and rendering these reconstructions along provided camera trajectories to track optimization progress. For each scene, we randomly sample the number of input images $\mathbf{C}$ from an exponential distribution, favoring smaller numbers of input images over larger ones. We track the optimization process by rendering the optimized 3D model every 100 steps across all provided camera trajectories. This results in a dataset comprising $(\mathbf{x}^c, \mathbf{c}^c, \mathbf{z}^t)$, where t ranges from 0 to 30,000 in increments of 100 steps. Each $\mathbf{z}^t$ is represented as rendered (latent) images at specific poses. During the training of the LVDM U-Net, we sample input conditioning images, $\mathbf{x}^{\mathrm{c}}$, and $\mathbf{z}^t$ (latent images rendered to poses $\mathbf{c}^g$), training the model to denoise $\mathbf{x}^{\mathrm{g}}$ at poses $\mathbf{c}^g$. This unrolled iterative training approach is substantially different from other models, which either train in a one-shot manner $p(\mathbf{x}^{\mathrm{g}} \mid \mathbf{c}^{\mathrm{c}}, \mathbf{x}^{\mathrm{c}})$ [18] or perform super-resolution of images [67].

Preventing Divergence. We noticed that during sampling, diffusion model can diverge and generate saturated, toyish-looking images. We speculate that this is due to conditioning becoming out-of-distribution than seen during training. Specifically, during training, model sees 3D reconstruction from sparse ground-truth images rather than model's own samples. To address this, we add Gaussian noise to conditioning images during both training and test time. This has the effect of bringing test and training distributions closer. Furthermore, we use classifier-free with a guidance scale < 1 to guide the samples towards 3D-unconditional model.

3.6 3D Representation and Rendering

Our sampling procedure aligns closely with standard 3D reconstruction methods, as outlined in Section 3. This allows us to leverage recent advancements in 3D scene representations and rendering techniques. We employ Gaussian splatting as the 3D representation [34], incorporating enhancements from [13]. For the loss function, we utilize both Mean Squared Error (MSE) and Learned Perceptual Image Patch Similarity (LPIPS).

Table 1: Quantitative comparison on 3D reconstruction. Our approach outperforms prior frameworks across nearly all metrics, except compared to CAT3D and ReconFusion, which are not publicly available and were evaluated on all categories of CO3D.

	1-view			6-view		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Ours	**16.23**	**0.423**	0.501	20.06	0.531	0.360
Ablation: one-shot	15.48	0.417	0.486	19.12	0.535	0.363
Ablation: MEO	15.39	0.404	0.601	17.98	0.371	0.540
Ablation: noisy conditioning	12.84	0.108	0.695	16.99	0.441	0.446
GSplatting [34]	14.89	0.399	0.504	19.41	0.298	0.637
GIBR†[2]	16.07	0.329	**0.456**	20.22	0.571	0.283
VD†[74]	13.18	0.144	0.714	-	-	-
PixelNeRF†[88]	15.06	0.278	0.615	16.86	0.366	0.545
RD †[3]	15.70	0.317	0.598	18.60	0.399	0.533
SparseFusion †[91]	12.06	-	0.630	-	-	-
ReconFusion* [86]	-	-	-	21.84	0.714	0.342
CAT3D* [18]	-	-	-	**22.79**	**0.726**	**0.292**
ZeroNVS* [65]	-	-	-	19.72	0.627	0.515
Zip-NeRF* [4]	-	-	-	14.48	0.497	0.617

4 Experiments

We conduct experiments on the CO3D dataset [61], which includes camera pose annotations, showing results on the hydrant class as in prior works [74]. Our evaluation centers on the task of sparse view 3D reconstruction, where we benchmark our method against several existing frameworks and ablations of our own approach.

In our experiments, we provide models with varying numbers of input frames—specifically 1 or 6 frames—to predict the 3D scene. The reconstructed scenes are then rendered from all viewpoints in the original video, and the quality of reconstruction is assessed by comparing the rendered images with ground-truth images. As metrics we use Peak Signal-to-Noise Ratio (PSNR), Learned Perceptual Image Patch Similarity (LPIPS), and Structural Similarity Index (SSIM). Since 3D reconstruction from sparse images is probabilistic in nature, we follow baseline works and draw multiple samples from the model, taking the best-performing sample. Our method is compared against state-of-the-art structure-in frameworks, including GIBR [2], RenderDiffusion [3], pixelNeRF [88], and SparseFusion [91]. Additionally, we evaluate our approach against one-shot generation methods, referencing results from [86, 18, 65]. Since these one-shot methods do not provide open-source code or compatible evaluation pipelines, we cite the numbers directly from their papers. As such, they are included primarily for broader context, and are not strictly comparable to our hydrant-only evaluation protocol. We further ablate our model to demonstrate the advantages of our iterative approach over one-shot methods, and highlighting the impact of autoregressive generation with Uncertainty-Guided Ordering (Sec. 3.2). Lastly, we compare our generative approach to the classical fitting of Gaussian splatting [34], underscoring that it fails in our sparse-view setting.

4.1 Sparse-View 3D Reconstruction

Table 1 presents quantitative results on sparse view 3D reconstruction. Each method is provided with N input images and their corresponding camera poses, and the reconstructed 3D representation is evaluated by rendering novel views. We follow prior work in considering two levels of difficulty: $N = 6$ and $N = 1$. Our model outperforms most prior approaches across key metrics. In the more challenging single-image 3D reconstruction task, as measured by PSNR and SSIM, our method surpasses the state-of-the-art GIBR [2], which is explicitly trained for novel view prediction and thus less general than ours. For 6-view 3D reconstruction, our method remains highly competitive, achieving strong performance across all three metrics. The main exceptions are GIBR, which performs slightly better, and CAT3D [18]. However, CAT3D is not publicly available, and its evaluation is based on all categories of CO3D rather than the more challenging outdoor hydrant class used in our benchmarks. Consequently, direct comparison may not fully reflect relative performance.

4.2 Ablations

We conduct ablation studies to analyze the key technical contributions of our framework, specifically iterative generation, 3D conditioning, and Uncertainty-Guided Ordering, demonstrating benefits of each in Table 1.

One-shot vs Iterative. During training, our probabilistic model learns to generate images at novel views given as input a conditioning image and its camera pose as well as a 3D representation. At test-time, we can use our generative model to generate a large dataset of images, similarly to one-shot approaches, such as CAT3D [18], ablating our iterative approach. In Table 1, we performed a quantitative ablation study, when the iterative approach is replaced with one-shot generation of the multi-view images, when other components are kept the same (e.g. Uncertainty-Guided Ordering). We see that such approach performs significantly worse than the iterative approach. Qualitatively, we see that images generated by the one-shot approach are highly inconsistent, in extreme cases, even changing the shape of the hydrant, which results in a blurry 3D reconstruction. In contrast, though our iterative approach may sample inconsistent results initially, it later converges on one particular 3D sample, leading to a sharp and detailed 3D reconstruction.

Ablating Conditioning Noising. To evaluate the impact of adding Gaussian noise to conditioning images, we conduct an ablation experiment with this component removed. Without noising, the conditioning images rendered from the 3D representation remain unaltered. Table 1 ("Ablation: noisy conditioning") shows that this leads to a noticeable drop in performance, suggesting that noising helps bridge the distribution gap between training and inference, leading to more stable and realistic generations.

Uncertainty-Guided Ordering. The query camera poses where new images are generated is sampled from $P_\lambda(\mathbf{c}^g \mid \mathbf{c}^c)$, which in our case, samples camera poses such that the overall entropy of generations is minimized (Sec. 3.2). To study the effect of such design, we ablate this component, replacing it with random sampling around the object (matching prior works). In Table 1, we show quantitative results ("Ablation: UGO"), observing that the model with Uncertainty-Guided Ordering achieves significantly better results across all metrics.

References

[1] T. Anciukevicius, P. Fox-Roberts, E. Rosten, and P. Henderson. Unsupervised causal generative understanding of images. *Advances in Neural Information Processing Systems*, 35:37037–37054, 2022.

[2] T. Anciukevičius, F. Manhardt, F. Tombari, and P. Henderson. Denoising diffusion via image-based rendering. In *The Twelfth International Conference on Learning Representations*, 2024.

[3] T. Anciukevičius, Z. Xu, M. Fisher, P. Henderson, H. Bilen, N. J. Mitra, and P. Guerrero. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12608–12618, June 2023.

[4] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *ICCV*, 2023.

[5] M. Á. Bautista, P. Guo, S. Abnar, W. Talbott, A. T. Toshev, Z. Chen, L. Dinh, S. Zhai, H. Goh, D. Ulbricht, A. Dehghan, and J. M. Susskind. GAUDI: A neural architect for immersive 3d scene generation. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[6] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, V. Jampani, and R. Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *CoRR*, abs/2311.15127, 2023.

[7] A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler, and K. Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[8] A. Cao, J. Johnson, A. Vedaldi, and D. Novotny. Lightplane: Highly-scalable components for neural 3d fields. *ArXiv*, 2024.

[9] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. D. Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

[10] E. R. Chan, K. Nagano, J. J. Park, M. Chan, A. W. Bergman, A. Levy, M. Aittala, S. D. Mello, T. Karras, and G. Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, October 2023.

[11] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[12] Y.-C. Cheng, H.-Y. Lee, S. Tuyakov, A. Schwing, and L. Gui. SDFusion: Multimodal 3d shape completion, reconstruction, and generation. In *CVPR*, 2023.

[13] F. Darmon, L. Porzi, S. Rota-Bulò, and P. Kontschieder. Robust gaussian splatting. *arXiv preprint arXiv:2404.04211*, 2024.

[14] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] Y. Deng, J. Yang, J. Xiang, and X. Tong. Gram: Generative radiance manifolds for 3d-aware image generation. In *IEEE Computer Vision and Pattern Recognition*, 2022.

[16] T. Devries, M. Á. Bautista, N. Srivastava, G. W. Taylor, and J. M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14284–14293, 2021.

[17] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.

[18] R. Gao, A. Holynski, P. Henzler, A. Brussee, R. Martin-Brualla, P. P. Srinivasan, J. T. Barron, and B. Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[20] J. Gu, Q. Gao, S. Zhai, B. Chen, L. Liu, and J. Susskind. Learning controllable 3d diffusion models from single-view images. *ArXiv*, 2023.

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

[22] J. Han, F. Kokkinos, and P. Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. In *European Conference on Computer Vision*, pages 333–350. Springer, 2025.

[23] P. Henderson and V. Ferrari. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *International Journal of Computer Vision (IJCV)*, 2019.

[24] P. Henderson, C. H. Lampert, and B. Bickel. Unsupervised video prediction from a single frame by estimating 3d dynamic scene structure. *CoRR*, abs/2106.09051, 2021.

[25] P. Henderson, V. Tsiminaki, and C. Lampert. Leveraging 2D data to learn textured 3D mesh generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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

[27] L. Höllein, A. Božič, N. Müller, D. Novotny, H.-Y. Tseng, C. Richardt, M. Zollhöfer, and M. Nießner. Viewdiff: 3d-consistent image generation with text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[28] H. Hu, Z. Zhou, V. Jampani, and S. Tulsiani. Mvd-fusion: Single-view 3d via depth-consistent multi-view generation. In *CVPR*, 2024.

[29] K.-H. Hui, R. Li, J. Hu, and C.-W. Fu. Neural wavelet-domain diffusion for 3d shape generation. In *SIGGRAPH Asia 2022 Conference Papers*, pages 1–9, 2022.

[30] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[31] A. Jain, M. Tancik, and P. Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021.

[32] J. T. Kajiya. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, page 143–150, New York, NY, USA, 1986. Association for Computing Machinery.

[33] A. Karnewar, A. Vedaldi, D. Novotny, and N. Mitra. Holodiffusion: Training a 3d diffusion model using 2d images. *ArXiv*, 2023.

[34] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

[35] S. W. Kim, B. Brown, K. Yin, K. Kreis, K. Schwarz, D. Li, R. Rombach, A. Torralba, and S. Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[36] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[37] X. Kong, S. Liu, X. Lyu, M. Taher, X. Qi, and A. J. Davison. Eschernet: A generative model for scalable view synthesis. *arXiv preprint arXiv:2402.03908*, 2024.

[38] A. R. Kosiorek, H. Strathmann, D. Zoran, P. Moreno, R. Schneider, S. Mokrá, and D. J. Rezende. Nerf-vae: A geometry aware 3d scene generative model. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5742–5752. PMLR, 2021.

[39] C. Lassner and M. Zollhofer. Pulsar: Efficient sphere-based neural rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1440–1449, 2021.

[40] M. Li, Y. Duan, J. Zhou, and J. Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[41] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin. Magic3d: High-resolution text-to-3d content creation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[42] M. Liu, C. Xu, H. Jin, L. Chen, M. Varma T, Z. Xu, and H. Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems*, 36, 2024.

[43] R. Liu, R. Wu, B. V. Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023.

[44] X. Liu, C. Zhou, and S. Huang. 3DGS-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[45] Y. Liu, C. Lin, Z. Zeng, X. Long, L. Liu, T. Komura, and W. Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] S. Luo and W. Hu. Diffusion probabilistic models for 3d point cloud generation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021.

[47] L. Melas-Kyriazi, I. Laina, C. Rupprecht, N. Neverova, A. Vedaldi, O. Gafni, and F. Kokkinos. Im-3d: Iterative multiview diffusion and reconstruction for high-quality 3d generation. *International Conference on Machine Learning, 2024*, 2024.

[48] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[49] Y. Mu, X. Zuo, C. Guo, Y. Wang, J. Lu, X. Wu, S. Xu, P. Dai, Y. Yan, and L. Cheng. Gsd: View-guided gaussian splatting diffusion for 3d reconstruction, 2024.

[50] N. Müller, Y. Siddiqui, L. Porzi, S. R. Bulo, P. Kontschieder, and M. Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4328–4338, 2023.

[51] T. Müller, A. Evans, C. Schied, and A. Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022.

[52] T. Nguyen-Phuoc, C. Li, L. Theis, C. Richardt, and Y.-L. Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019.

[53] T. Nguyen-Phuoc, C. Richardt, L. Mai, Y.-L. Yang, and N. Mitra. Blockgan: Learning 3d object-aware scene representations from unlabelled images. In *Advances in Neural Information Processing Systems 33*, Nov 2020.

[54] M. Niemeyer, J. T. Barron, B. Mildenhall, M. S. M. Sajjadi, A. Geiger, and N. Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2022.

[55] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob. Mitsuba 2: A retargetable forward and inverse renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 38(6), Dec. 2019.

[56] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. Featured Certification.

[57] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, A. Vyas, B. Shi, C.-Y. Ma, C.-Y. Chuang, D. Yan, D. Choudhary, D. Wang, G. Sethi, G. Pang, H. Ma, I. Misra, J. Hou, J. Wang, K. Jagadeesh, K. Li, L. Zhang, M. Singh, M. Williamson, M. Le, M. Yu, M. K. Singh, P. Zhang, P. Vajda, Q. Duval, R. Girdhar, R. Sumbaly, S. S. Rambhatla, S. Tsai, S. Azadi, S. Datta, S. Chen, S. Bell, S. Ramaswamy, S. Sheynin, S. Bhattacharya, S. Motwani, T. Xu, T. Li, T. Hou, W.-N. Hsu, X. Yin, X. Dai, Y. Taigman, Y. Luo, Y.-C. Liu, Y.-C. Wu, Y. Zhao, Y. Kirstain, Z. He, Z. He, A. Pumarola, A. Thabet, A. Sanakoyeu, A. Mallya, B. Guo, B. Araya, B. Kerr, C. Wood, C. Liu, C. Peng, D. Vengertsev, E. Schonfeld, E. Blanchard, F. Juefei-Xu, F. Nord, J. Liang, J. Hoffman, J. Kohler, K. Fire, K. Sivakumar, L. Chen, L. Yu, L. Gao, M. Georgopoulos, R. Moritz, S. K. Sampson, S. Li, S. Parmeggiani, S. Fine, T. Fowler, V. Petrovic, and Y. Du. Movie gen: A cast of media foundation models, 2024.

[58] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023.

[59] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[60] A. Radford and K. Narasimhan. Improving language understanding by generative pre-training. OpenAI, 2018.

[61] J. Reizenstein, R. Shapovalov, P. Henzler, L. Sbordone, P. Labatut, and D. Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

[62] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[63] B. Roessle, N. Müller, L. Porzi, S. R. Bulò, P. Kontschieder, and M. Nießner. Ganerf: Leveraging discriminators to optimize neural radiance fields. *ACM Trans. Graph.*, 42(6), nov 2023.

[64] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, June 2022.

[65] K. Sargent, Z. Li, T. Shah, C. Herrmann, H.-X. Yu, Y. Zhang, E. R. Chan, D. Lagun, L. Fei-Fei, D. Sun, and J. Wu. ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. *CVPR, 2024*, 2023.

[66] K. Schwarz, Y. Liao, M. Niemeyer, and A. Geiger. GRAF: generative radiance fields for 3d-aware image synthesis. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[67] Y. Shen, D. Ceylan, P. Guerrero, Z. Xu, N. J. Mitra, S. Wang, and A. Frühstück. Supergaussian: Repurposing video models for 3d super resolution. In *European Conference on Computer Vision*, pages 215–233. Springer, 2025.

[68] J. R. Shue, E. R. Chan, R. Po, Z. Ankner, J. Wu, and G. Wetzstein. 3d neural field generation using triplane diffusion. *arXiv preprint arXiv:2211.16677*, 2022.

[69] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[70] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

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

[72] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[73] R. S. Sutton. The bitter lesson. `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`, 2019.

[74] S. Szymanowicz, C. Rupprecht, and A. Vedaldi. Viewset diffusion: (0-)image-conditioned 3D generative models from 2D data. In *ICCV*, 2023.

[75] J. Tang, Z. Chen, X. Chen, T. Wang, G. Zeng, and Z. Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

[76] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *The Twelfth International Conference on Learning Representations*, 2024.

[77] S. Tang, J. Chen, D. Wang, C. Tang, F. Zhang, Y. Fan, V. Chandra, Y. Furukawa, and R. Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. *arXiv preprint arXiv:2402.12712*, 2024.

[78] A. Tewari, T. Yin, G. Cazenavette, S. Rezchikov, J. B. Tenenbaum, F. Durand, W. T. Freeman, and V. Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[79] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

[80] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[81] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[82] V. Voleti, C.-H. Yao, M. Boss, A. Letts, D. Pankratz, D. Tochilkin, C. Laforte, R. Rombach, and V. Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*, pages 439–457. Springer, 2025.

[83] H. Wang, X. Du, J. Li, R. A. Yeh, and G. Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023.

[84] T. Wang, B. Zhang, T. Zhang, S. Gu, J. Bao, T. Baltrusaitis, J. Shen, D. Chen, F. Wen, Q. Chen, and B. Guo. Rodin: A generative model for sculpting 3d digital avatars using diffusion. *ArXiv*, 2022.

[85] Z. Wang, C. Lu, Y. Wang, F. Bao, C. Li, H. Su, and J. Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

[86] R. Wu, B. Mildenhall, P. Henzler, K. Park, R. Gao, D. Watson, P. P. Srinivasan, D. Verbin, J. T. Barron, B. Poole, and A. Holynski. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, pages 21551–21561, 2024.

[87] Y. Xu, H. Tan, F. Luan, S. Bi, P. Wang, J. Li, Z. Shi, K. Sunkavalli, G. Wetzstein, Z. Xu, and K. Zhang. DMV3d: Denoising multi-view diffusion using 3d large reconstruction model. In *The Twelfth International Conference on Learning Representations*, 2024.

[88] A. Yu, V. Ye, M. Tancik, and A. Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021.

[89] B. Zhang, Y. Chen, C. Wang, F. Zhao, Y. Tang, D. Chen, and B. Guo. Gaussiancube: Structuring gaussian splatting using optimal transport for 3d generative modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[90] J. J. Zhou, H. Gao, V. Voleti, A. Vasishta, C.-H. Yao, M. Boss, P. Torr, C. Rupprecht, and V. Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint*, 2025.

[91] Z. Zhou and S. Tulsiani. Sparsefusion: Distilling view-conditioned diffusion for 3d reconstruction. In *CVPR*, 2023.

Supplementary Material

We have introduced a novel probabilistic framework for 3D reconstruction that uses autoregressive image generation conditioned on a iteratively updated 3D representation. By iteratively sampling images consistent with the 3D representation, our approach overcomes the limitations of prior 2D and 3D generative models at sampling many images, enabling state-of-the-art single-image 3D reconstructions of unbounded scenes at arbitrary resolutions. In this section, we discuss limitations (A)), additional details on the generative model (B)), additional results (C), 3D representation (D) and pose sampling (E) used in our framework.

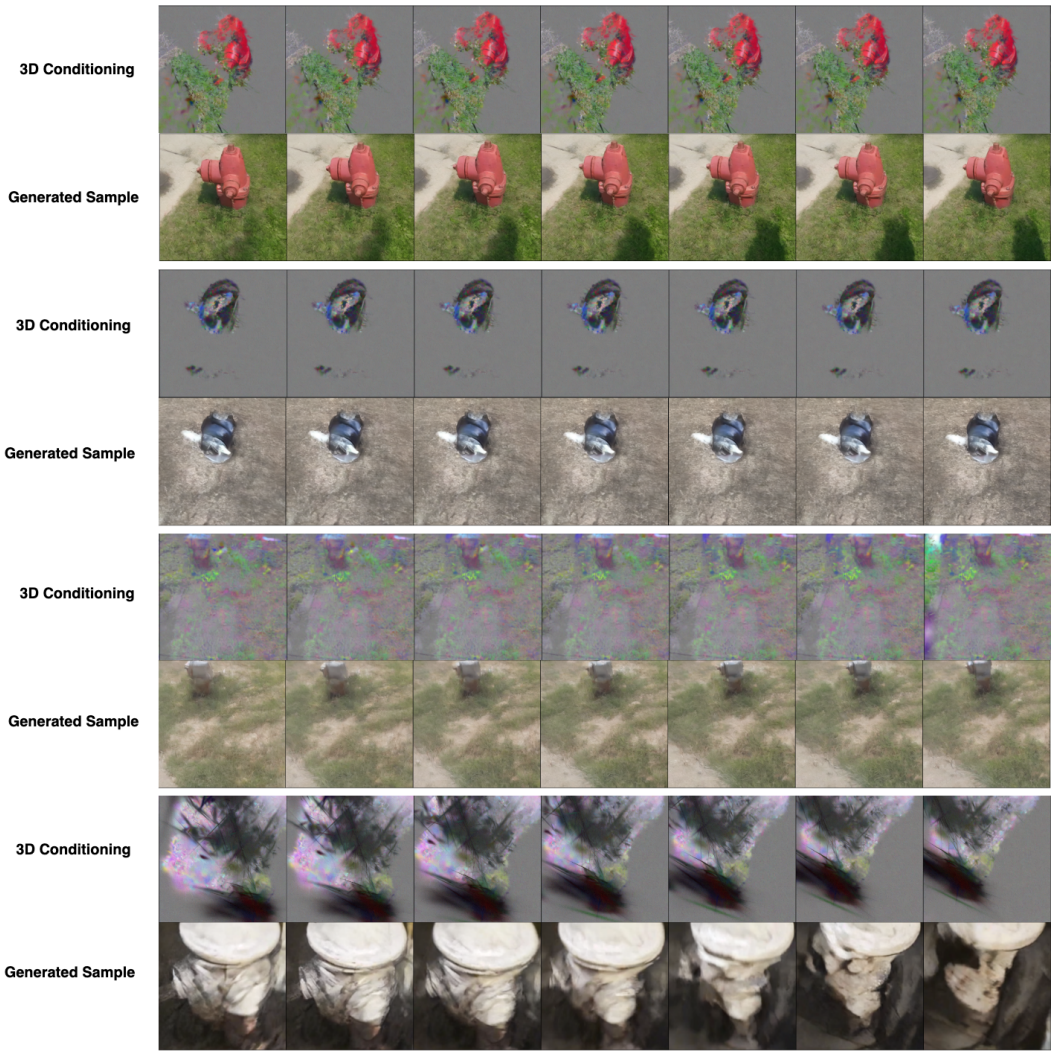

Figure 3: Visualization of a training step in our generative model. The model denoises noisy latent frames while conditioned on both input images and the 3D representation via rendered views. The process demonstrates the model's ability to generate content that is structurally consistent with the evolving 3D representation, effectively filling in missing details.

A Limitations.

Our model generates 3D scenes iteratively, which enhances the fidelity of the 3D representation. However, this approach incurs a trade-off in terms of generation speed, as batches must be repeatedly generated by the generative model. Future work could explore the use of faster generative models than diffusion models or investigate one-step prediction techniques to mitigate this issue. Another

limitation is that the transition from images $\mathbf{x}$ to 3D representation $\mathbf{z}$ occurs without priors. While this allows for a clear separation between reconstruction method and 3D representation, it also means that the 3D reconstruction process lacks inherent priors which often results in artifacts that are typically not present in 3D-aware methods. Future research could address this by incorporating 3D-aware models, to directly update 3D representation $\mathbf{z}^i$.

B Additional details on Generative Model

As described in Section 3.1, we train a latent video diffusion model conditioned on both the 3D representation and the input image. During training, the model denoises sequences of 20 frames, where 2D images – rendered from the 3D model – are concatenated along the channel dimension with noisy latents. These are fed into the denoising UNet. The input images are processed using CLIP and DINO feature extractors, and at each block of the UNet, the denoising UNet attends to these features via cross-attention.

To enhance robustness, we apply random conditioning dropout during training: input images are dropped with a probability of 0.1, 3D conditioning is dropped with a probability of 0.1, and both are simultaneously dropped with a probability of 0.1. During sampling, we use classifier-free guidance to control the trade-off between fidelity and diversity.

C Additional Qualitative Results

We visualize random training steps in Figure 3, where the model is tasked with denoising noisy latents while conditioned on both input images and the 3D representation. The visualization highlights how the model learns to predict content that is structurally consistent with the 3D representation, successfully inferring missing details.

D Additional Details on 3D Representation and Rendering

Though our framework is independent of 3D representation, for all our experiments, we use 3D Gaussian Splatting [34] as the underlying scene representation. The 3D Gaussians are initialized using a depth estimator applied to the generated video frames, extracting a 3D point cloud. This approach aligns with prior methods such as CAT3D [18], ensuring a structured and scalable representation of the scene.

E Camera Pose Sampling

The camera poses where new images are being generated are sampled from $P_\lambda(\mathbf{c}^g \mid \mathbf{c}^c)$ which we carefully design so that that P_θ can easily generate consistent images. In practice, this involves: (i) Sampling views near previously generated ones rather than those on the opposite side of the input image, as the latter introduces high uncertainty due to unobserved regions. This is done autoregressively, using the previous endpoint as the new starting point. (ii) Sampling camera trajectories that resemble those in the CO3D dataset, which the model was trained on. For the latter, we also create zoom-out and zoom-in effects, by estimating world center and adding or subtracting small amounts of distances to each camera pose.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: WRITE HERE

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: At the end of the paper.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: No theoretical results

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Details provided in the method and supplementary

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Will be made available post publication.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Provided in method, supplementary and post publication release.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification:

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes in supplementary

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We do not foresee any direct societal impact resulting from this work. While the proposed method involves image generation and 3D reconstruction, it is not designed for or applied to human subjects. Although, in theory, similar techniques could be adapted for misuse (e.g. in generating deepfakes), our work does not directly enable or target such applications.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: See above

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite all used datasets and followed their licence.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.