GAMBA: MARRY GAUSSIAN SPLATTING WITH MAMBA FOR SINGLE-VIEW 3D RECONSTRUCTION

Qiuhong Shen[1*] **Xuanyu Yi**[3*] **Zike Wu**[3*] **Pan Zhou**[2,4†] **Hanwang Zhang**[3,5]
Shuicheng Yan[5] **Xinchao Wang**[1†]
[1]National University of Singapore [2]Singapore Management University
[3]Nanyang Technological University [4]Sea AI Lab [5]Skywork AI

ABSTRACT

We tackle the challenge of efficiently reconstructing a 3D asset from a *single* image with growing demands for automated 3D content creation pipelines. Previous methods primarily rely on Score Distillation Sampling (SDS) and Neural Radiance Fields (NeRF). Despite their significant success, these approaches encounter practical limitations due to lengthy optimization and considerable memory usage. In this report, we introduce Gamba, an end-to-end amortized 3D reconstruction model from single-view images, emphasizing two main insights: (1) 3D representation: leveraging a *large number* of 3D Gaussians for an efficient 3D Gaussian splatting process; (2) Backbone design: introducing a Mamba-based sequential network that facilitates context-dependent reasoning and linear scalability with the sequence (token) length, accommodating a substantial number of Gaussians. Gamba incorporates significant advancements in data preprocessing, regularization design, and training methodologies. We assessed Gamba against existing optimization-based and feed-forward 3D generation approaches using the real-world scanned OmniObject3D dataset. Here, Gamba demonstrates competitive generation capabilities, both qualitatively and quantitatively, while achieving remarkable speed, approximately 0.6 second on a single NVIDIA A100 GPU.

1 INTRODUCTION

We tackle the challenge of efficiently extracting a 3D asset from a single image, an endeavor with substantial implications across diverse industrial sectors. This endeavor facilitates AR/VR content generation from a single snapshot and aids in the development of autonomous vehicle path planning through monocular perception Sun et al. (2023); Gul et al. (2019); Yi et al. (2023).

Previous approaches to single-view 3D reconstruction have mainly been achieved through Score Distillation Sampling (SDS) Poole et al. (2022), which leverages pre-trained 2D diffusion models Graikos et al. (2022); Rombach et al. (2022) to guide optimization of the underlying representations of 3D assets. These optimization-based approaches have achieved remarkable success, known for their high-fidelity and generalizability. However, they require a time-consuming per-instance optimization process Tang (2022); Wang et al. (2023d); Wu et al. (2024) to generate a single object and also suffer from artifacts such as the "multi-face" problem arising from bias in pre-trained 2D diffusion models Hong et al. (2023a). On the other hand, previous approaches predominantly utilized neural radiance fields (NeRF) Mildenhall et al. (2021); Barron et al. (2021), which are equipped with high-dimensional multi-layer perception (MLP) and inefficient volume rendering Mildenhall et al. (2021). This computational complexity significantly limits practical applications on limited compute budgets. For instance, the Large reconstruction Model (LRM) Hong et al. (2023b) is confined to a resolution of 32 using a triplane-NeRF Shue et al. (2023) representation, and the resolution of renderings is limited to 128 due to the bottleneck of online volume rendering.

To address these challenges and thus achieve *efficient* single-view 3D reconstruction, we are seeking an amortized generative framework with the groundbreaking 3D Gaussian Splatting, notable

[*]Equal Contribution
[†]Corresponding author : Xinchao Wang (xinchao@nus.edu.sg) and Pan Zhou (panzhou@smu.edu.sg)
[‡]Work in progress, partially done in Sea AI Lab and 2050 Research, Skywork AI

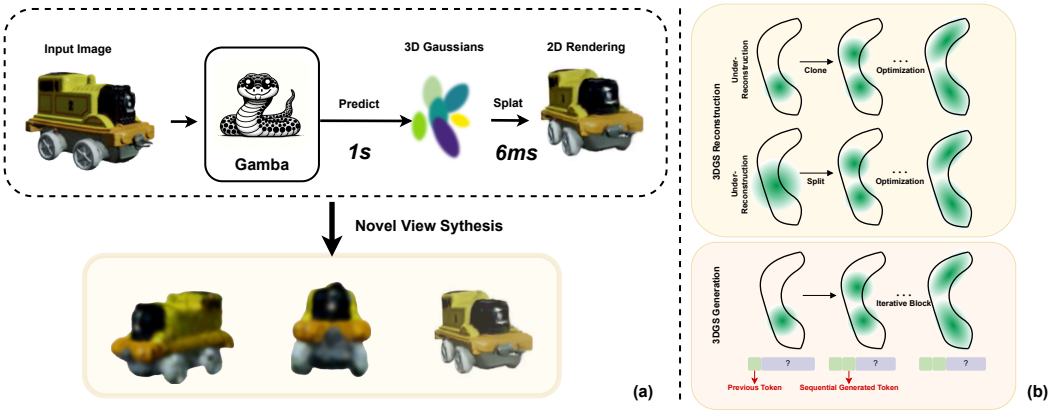

Figure 1: (a): We propose Gamba, an end-to-end, feed-forward single-view reconstruction pipeline, which marries 3D Gaussian Splatting with Mamba to achieve fast reconstruction. (b): The relationship between the 3DGS generation process and the Mamba sequential predicting pattern.

for its memory-efficient and high-fidelity tiled rendering Kerbl et al. (2023); Zwicker et al. (2002); Chen & Wang (2024); Wang et al. (2024). Despite recent exciting progress Tang et al. (2023), how to properly and immediately generate 3D Gaussians remains a less studied topic. Recent prevalent 3D amortized generative models Hong et al. (2023b); Wang et al. (2023b); Xu et al. (2024; 2023); Zou et al. (2023); Li et al. (2023) predominantly use transformer-based architecture as their backbones Vaswani et al. (2017); Peebles & Xie (2023), but we argue that these widely used architectures are sub-optimal for generating 3DGS. The crucial challenge stems from the fact that 3DGS requires a sufficient number of 3D Gaussians to accurately represent a 3D model or scene. However, the spatio-temporal complexity of Transformers increases quadratic-ally with the number of tokens Vaswani et al. (2017), which limits the expressiveness of the 3DGS due to the insufficient token counts for 3D Gaussians. Furthermore, the 3DGS parameters possess specific physical meanings, making the simultaneous generation of 3DGS parameters a more challenging task.

To tackle the above challenges, we start by revisiting the 3DGS reconstruction process from multi-view images. The analysis presented in fig 1(b) reveals that 3DGS densification during the reconstruction process can be conceptualized as a sequential generation based on previously generated tokens. With this insight, we introduce a novel architecture for *end-to-end* 3DGS generation dubbed Gaussian Mamba (Gamba), which is built upon a new scalable sequential network, Mamba Gu & Dao (2023a). Our Gamba enables context-dependent reasoning and scales linearly with sequence (token) length, allowing it to efficiently mimic the inherent process of 3DGS reconstruction when generating 3D assets enriched with a sufficient number of 3D Gaussians. Due to its feed-forward architecture and efficient rendering, Gamba is exceptionally fast, requiring only about 1 seconds to generate a 3D asset and 6 ms for novel view synthesis, which is 5000× faster than previous optimization-based methods Wu et al. (2024); Weng et al. (2023); Qian et al. (2023) while achieving comparable generation quality.

We demonstrate the superiority of Gamba on the OmniObject3D dataset Wu et al. (2023). Both qualitative and quantitative experiments clearly indicate that Gamba can instantly generate high-quality and diverse 3D assets from a single image, continuously outperforming other state-of-the-art methods. In summary, we make three-fold contributions:

- We introduce GambaFormer, a simple state space model to process 3D Gaussian Splatting, which has global receptive fields with linear complexity.
- Integrated with the GambaFormer, we present Gamba, an amortized, end-to-end 3D Gaussian Splatting Generation pipeline for fast and high-quality single-view reconstruction.
- Extensive experiments show that Gamba outperforms the state-of-the-art baselines in terms of reconstruction quality and speed.

2 RELATED WORKS

Amortized 3D Generation. Amortized 3D generation is able to instantly generate 3D assets in a feed-forward manner after training on large-scale 3D datasets Wu et al. (2023); Deitke et al. (2023); Yu et al. (2023), in contrast to tedious SDS-based optimization methods Wu et al. (2024); Lin et al. (2023); Weng et al. (2023); Guo et al. (2023); Tang (2022). Previous works Nichol et al. (2022); Nash et al. (2020) married de-noising diffusion models with various 3D explicit representations (*e.g.*, point cloud and mesh), which suffers from lack of generalizablity and low texture quality. Recently, pioneered by LRM Hong et al. (2023b), several works utilize the capacity and scalability of the transformer Peebles & Xie (2023) and propose a full transformer-based regression model to decode a NeRF representation from triplane features. The following works extend LRM to predict multi-view images Li et al. (2023), combine with diffusion Xu et al. (2023), and pose estimation Wang et al. (2023b). However, their triplane NeRF-based representation is restricted to inefficient volume rendering and relatively low resolution with blurred textures. Gamba instead seeks to train an efficient feed-forward model marrying Gaussian splatting with Mamba for single-view 3D reconstruction.

Gaussian Splatting for 3D Generation. The explicit nature of 3DGS facilitates real-time rendering capabilities and unprecedented levels of control and editability, making it highly relevant for 3D generation. Several works have effectively utilized 3DGS in conjunction with optimization-based 3D generation Wu et al. (2024); Poole et al. (2022); Lin et al. (2023). For example, DreamGaussian Tang et al. (2023) utilizes 3D Gaussian as an efficient 3D representation that supports real-time high-resolution rendering via rasterization. Despite the acceleration achieved, generating high-fidelity 3D Gaussians using such optimization-based methods still requires several minutes and a large computational memory demand. TriplaneGaussian Zou et al. (2023) extends the LRM architecture with a hybrid triplane-Gaussian representation. AGG Xu et al. (2024) decomposes the geometry and texture generation task to produce coarse 3D Gaussians, further improving its fidelity through Gaussian Super Resolution. Splatter image Szymanowicz et al. (2023) and PixelSplat Charatan et al. (2023) propose to predict 3D Gaussians as pixels on the output feature map of two-view images. LGM Tang et al. (2024) generates high-resolution 3D Gaussians by fusing information from multi-view images generated by existing multi-view diffusion models Shi et al. (2023); Wang & Shi (2023) with an asymmetric U-Net. Among them, our Gamba demonstrates its superiority and structural elegance with *single image* as input and an *end-to-end*, *single-stage*, feed-forward manner.

State Space Models. Utilizing ideas from the control theory Glasser (1985), the integration of linear state space equations with deep learning has been widely employed to tackle the modeling of sequential data. The promising property of linearly scaling with sequence length in long-range dependency modeling has attracted great interest from searchers. Pioneered by LSSL Gu et al. (2021b) and S4 Gu et al. (2021a), which utilize linear state space equations for sequence data modeling, follow-up works mainly focus on memory efficiency Gu et al. (2021a), fast training speed Gu et al. (2022b;a) and better performance Mehta et al. (2022); Wang et al. (2023a). More recently, Mamba Gu & Dao (2023b) integrates a selective mechanism and efficient hardware design, outperforms Transformers Vaswani et al. (2017) on natural language and enjoys linear scaling with input length. Building on the success of Mamba, Vision Mamba Zhu et al. (2024) and VMamba Liu et al. (2024) leverage the bidirectional Vim Block and the Cross-Scan Module respectively to gain data-dependent global visual context for visual representation; U-Mamba Ma et al. (2024) and Vm-unet Ruan & Xiang (2024) further bring Mamba into the field of medical image segmentation. PointMamba Liang et al. (2024a) and Point Cloud Mamba Zhang et al. (2024) adapt Mamba for point cloud understanding through reordering and serialization strategy. In this manuscript, we explore the capabilities of Mamba in single-view 3D reconstruction and introduce Gamba.

3 PRELIMINARY

3.1 3D GAUSSIAN SPLATTING

3D Gaussian Splatting (3DGS) Kerbl et al. (2023) has gained prominence as an efficient explicit 3D representation, using anisotropic 3D Gaussians to achieve intricate modeling. Each Gaussian, denoted as G, is defined by its mean $\mu \in \mathbb{R}^3$, covariance matrix Σ, associated color $c \in \mathbb{R}^3$, and opacity $\alpha \in \mathbb{R}$. To be better optimized, the covariance matrix Σ is constructed from a scaling matrix

$S \in \mathbb{R}^3$ and a rotation matrix $R \in \mathbb{R}^{3\times3}$ as follows:

$$\Sigma = RSS^TR^T. \tag{1}$$

This formulation allows for the optimization of Gaussian parameters separately while ensuring that Σ remains positive semi-definite. A Gaussian with mean μ is defined as follows:

$$G(x) = \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right), \tag{2}$$

where x represents the offset from μ to a given point x. In the blending phase, the color accumulation C is calculated by:

$$C = \sum_{i\in N} c_i\alpha_i G(x_i)\prod_{j=1}^{i-1}\left(1-\alpha_j G(x_j)\right). \tag{3}$$

3DGS utilizes a tile-based rasterizer to facilitate real-time rendering and integrates Gaussian parameter optimization with a dynamic density control strategy. This approach allows for the modulation of Gaussian counts through both densification and pruning operations.

3.2 State Space Models

State Space Models (SSMs) Gu et al. (2021a) have emerged as a powerful tool for modeling and analyzing complex physical systems, particularly those that exhibit linear time-invariant (LTI) behavior. The core idea behind SSMs is to represent a system using a set of first-order differential equations that capture the dynamics of the system's state variables. This representation allows for a concise and intuitive description of the system's behavior, making SSMs well-suited for a wide range of applications. The general form of an SSM can be expressed as follows:

$$\begin{aligned} \dot{h}(t) &= Ah(t) + Bx(t), \\ y(t) &= Ch(t) + Dx(t). \end{aligned} \tag{4}$$

where $h(t)$ denotes the state vector of the system at time t, while $\dot{h}(t)$ denotes its time derivative. The matrices A, B, C, and D encode the relationships between the state vector, the input signal $x(t)$, and the output signal $y(t)$. These matrices play a crucial role in determining the system's response to various inputs and its overall behavior.

One of the challenges in applying SSMs to real-world problems is that they are designed to operate on continuous-time signals, whereas many practical applications involve discrete-time data. To bridge this gap, it is necessary to discretize the SSM, converting it from a continuous-time representation to a discrete-time one. The discretized form of an SSM can be written as:

$$\begin{aligned} h_k &= \bar{A}h_{k-1} + \bar{B}x_k, \\ y_k &= \bar{C}h_k + \bar{D}x_k. \end{aligned} \tag{5}$$

Here, k represents the discrete time step, and the matrices $\bar{A}$, $\bar{B}$, $\bar{C}$, and $\bar{D}$ are the discretized counterparts of their continuous-time equivalents. The discretization process involves sampling the continuous-time input signal $x(t)$ at regular intervals, with a sampling period of Δ. This leads to the following relationships between the continuous-time and discrete-time matrices:

$$\begin{aligned} \bar{A} &= (I - \Delta/2\cdot A)^{-1}(I + \Delta/2\cdot A), \\ \bar{B} &= (I - \Delta/2\cdot A)^{-1}\Delta B, \\ \bar{C} &= C. \end{aligned} \tag{6}$$

Selective State Space Models Gu & Dao (2023a) are proposed to address the limitations of traditional SSMs in adapting to varying input sequences and capturing complex, input-dependent dynamics. The key innovation in Selective SSMs is the introduction of a selection mechanism that allows the model to efficiently select data in an input-dependent manner, enabling it to focus on relevant information and ignore irrelevant inputs. The selection mechanism is implemented by parameterizing the SSM matrices $\bar{B}$, $\bar{C}$, and Δ based on the input x_k. This allows the model to dynamically adjust its behavior depending on the input sequence, effectively filtering out irrelevant information and remembering relevant information indefinitely.

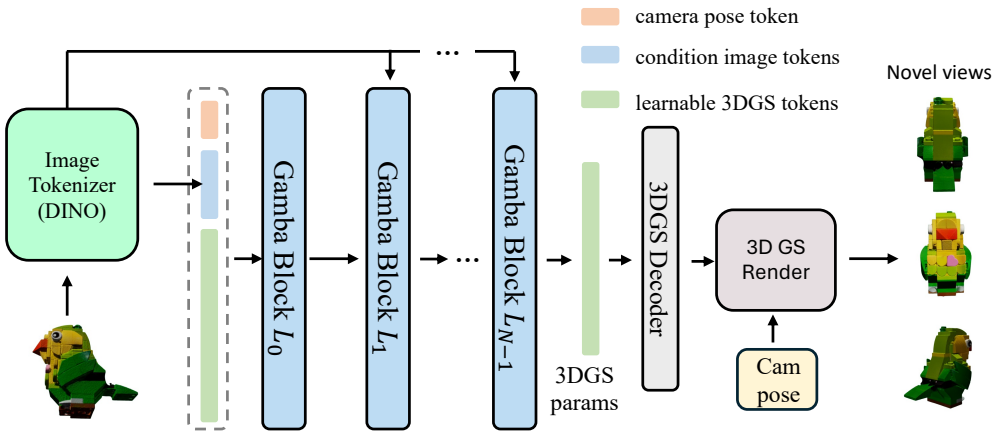

Figure 2: Overall architecture of Gamba.

4 METHOD

In this section, we detail our proposed single-view 3D reconstruction pipeline with 3D Gaussian Splatting (Fig. 2), called "Gamba", whose core mechanism is the GambaFormer to predict 3D Gaussian from a single input image (Section 4.2). We design an elaborate Gaussian parameter constrain robust training pipeline (Section 4.3) to ensure stability and high quality.

4.1 OVERALL TRAINING PIPELINE

Given a set of multi-view images and their corresponding camera pose pairs $\{\mathbf{x}, \pi\}$ of an object, Gamba first transforms the reference image $\mathbf{x}_{\text{ref}}$ and pose π_{ref} into learnable tokens, which are then concatenated with the initial positional embedding to predict a set of 3D Gaussians. Subsequently, the predicted 3D Gaussians are rendered into 2D multi-view images with differentiable renderer Kerbl et al. (2023), which are directly supervised by the provided ground-truth images $\{\mathbf{x}, \pi\}$ at both reference and novel poses through image-space reconstruction loss.

4.2 GAMBAFORMER

Image Tokenizer. Parallel to previous work Hong et al. (2023b), the reference RGB image $\mathbf{x} \in \mathbb{R}^{c\times H\times W}$ is tokenized with a visual transformer (ViT) model DINO v2 Oquab et al. (2023), which has demonstrated robust feature extraction capability through self-supervised pre-training, to extract both semantic and spatial representations of a sequence of tokens $X \in \mathbb{R}^{L\times C}$, with a length of L and channel dimensions as C.

Camera Condition. As the camera poses π_{ref} of the reference images vary across sampled 3D objects in training phase, we need to embed the camera features as a condition for our GambaFormer. We construct the camera matrix with the 12 parameters containing the rotation and translation of the camera extrinsic and 4 parameters $[fx, fy, cx, cy]$ denoting the camera intrinsic, which are further transformed into a high-dimensional camera embedding T with a multi-layer perceptron (MLP). Note that Gamba does not depend on any canonical pose, and the ground truth π is only applied as input during training for multi-view supervision.

Predicting from 3DGS tokens. In the GambaFormer architecture, as shown in Fig.2, inputs are segmented into three distinct segments: the camera embedding T, reference image tokens X, and a set of learnable 3DGS embeddings $E \in \mathbb{R}^{N\times D}$, with N representing the total number of 3DGS tokens, typically set to $N = 16384$, and D is the hidden dimension of each Gamba blocks.

Details of the Gamba block are shown in Fig. 3, alongside the original Mamba block architecture. The Mamba block is capable of efficiently processing long sequences of tokens; we note that the current Mamba variants Liu et al. (2024); Zhu et al. (2024); Liang et al. (2024b) do not involve traditional cross-attentions in their methods. Therefore, the use of cross-attention with Mamba re-

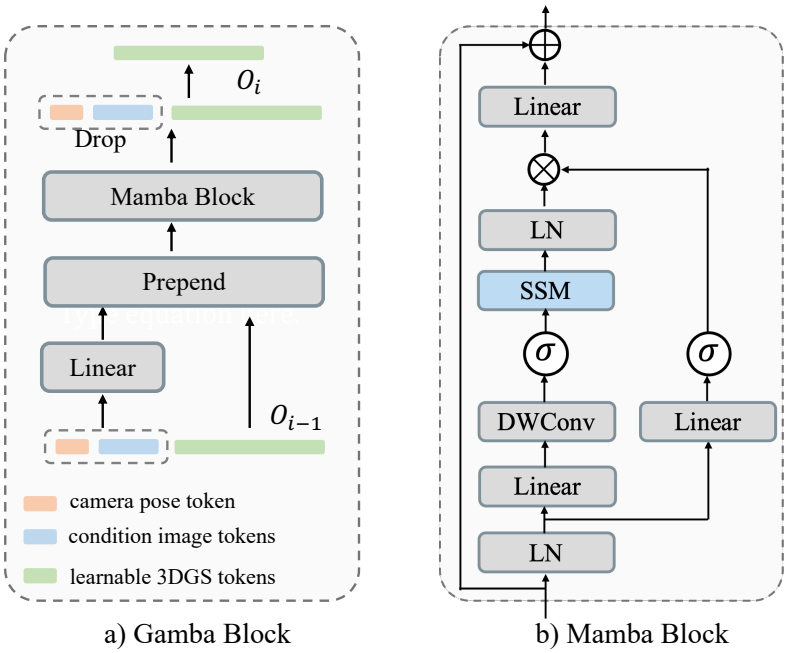

Figure 3: Single Gamba block, where layer normalization (LN), SSM, depth-wise convolution Chollet (2017), and residual connections are employed.

mains unexplored. Given the inherent unidirectional scan order of Mamba, we propose using this characteristic to achieve conditional prediction. Specifically, the Gamba block consists of a single Mamba block, two linear projections, and simple prepend operations:

$$\begin{aligned} H_i &= M_i(\text{Prepend}(P_c T, P_x X), O_{i-1}) \\ O_i &= \text{Drop}(H_i, \text{Index}(P_c T, P_x X)) \end{aligned} \tag{7}$$

where $P_c \in \mathbb{R}^{D\times 16}$ and $P_x \in \mathbb{R}^{D\times C}$ denote the learnable camera projection and image token projection, respectively. Prepend refers to the operation of prepending projected camera embeddings and image tokens before the hidden 3DGS token features in each Gamba layer. Drop refers to the operation of removing the previously prepended tokens from the output H_i of the Mamba block M_i, based on their index.

Gaussian Decoder. With stacked Gamba layers, our GambaFormer is capable of retrieving hidden features for each 3DGS token associated with the target object. Subsequently, it predicts the details of 3D Gaussian Splatting parameters using a dedicated Gaussian Decoder. The Gaussian Decoder initially processes the output from the GambaFormer, employing several MLP layers in a feed-forward manner. Then it uses linear layers to separately predict the attributes of each 3D Gaussian G_j: the center position $m_j \in \mathbb{R}^3$, opacity $o_j \in \mathbb{R}$, and color $c_j \in \mathbb{R}^{12}$, given the adoption of first-order spherical harmonics. The position m_j is predicted as a discrete probability distribution and is bound within the range of $[-1, 1]$ analogously to previous work in objection detection Duan et al. (2019).

4.3 Robust Amortized Training.

Gaussian Parameter Constraint. Parallel to AGG Xu et al. (2024), our amortized framework involves training a generator to concurrently produce 3D Gaussians for a broad array of objects from various categories, diverging from the traditional 3D Gaussian splatting approach where 3D Gaussians are individually optimized for each object. Therefore, we adopt the Gaussian parameter constraint mechanism to fit our feed-forward setting. Specifically, we use a fixed number of 3D Gaussians since the Mamba blocks only support a fixed token length. For the constraint on the predicted positions, our approach leverages a novel 2D Gaussian constraint instead of directly utilizing point clouds from the 3D surface for separating Zou et al. (2023) positions of 3D Gaussian or warm-

up Xu et al. (2024), whose point cloud constraint largely limits the scalability of the pre-trained 3D reconstruction model.

Our key insight is that, though the given multi-view image sets cannot depict the accurate 3D surface, it can define a rough 3D range of the object. Based on this insight, we devise a more scalable approach for Gauusian position constraint using only multi-view image sets in training. Specifically, as illustrated in Fig. 4, when the projected 2D Gaussian center is outside the object's contours, we impose an explicit constraint to pull it inside. This constraint is formulated as minimizing the distance between the 2D Gaussian center and the corresponding contour position at the same radial angles. For a fast approximation of each 2D Gaussian's contour position, object masks are discretized into radial polygons by ray casting, then followed by linear interpolation between continuous angles. By explicitly constraining projected 2D Gaussians with multi-view image sets, the Gaussian Decoder can quickly converge to predict rough 3D shapes.

Data Augmentation. Gamba primarily emphasizes reconstructing foreground objects, rather than modeling the background. Although our training data consist of single images with pure color backgrounds, we have observed that the 2D renderings during inference often present cluttered backgrounds. To avoid over-fitting this pattern, we implement a random color strategy for background generation. Furthermore, we employ a semantic-aware filter Yi et al. (2022) based on the CLIP similarity metric Radford et al. (2021), to select the canonical pose as the reference view for training stability and faster convergence.

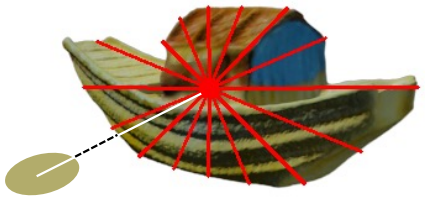

Figure 4: **Radial polygon mask**. Object masks are divided into polygon masks by 2D ray casting from the image center to the contours.

Training Objective. Taking advantage of the efficient tiled rasterizer Kerbl et al. (2023) for 3D Gaussians, Gamba is trained in an end-to-end manner with image-space reconstruction loss at both reference and novel views. Our final objective comprises four key terms:

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{mask}} \cdot \mathcal{L}_{\text{mask}} + \lambda_{\text{lpips}} \cdot \mathcal{L}_{\text{lpips}} + \lambda_{\text{dist}} \cdot \mathcal{L}_{\text{dist}}\,, \tag{8}$$

where $\mathcal{L}_{\text{rgb}}$ is the mean square error (MSE) loss in RGB space; $\mathcal{L}_{\text{mask}}$ and $\mathcal{L}_{\text{lpips}}$ refer to the alpha mask loss and the VGG-based LPIPS loss Zhang et al. (2018), respectively. The last term, L_{dist}, imposes an explicit constraint on the projected 2D Gaussians, also implemented as MSE, which is only adopted during initial training as this term converges exceptionally quickly.

5 Experiments

5.1 Implementation Details

Datasets. We used a filtered OmniObject3D Wu et al. (2023) dataset for Gamba pre-training, which contains high-quality scans of real-world 3D objects. Following AGG Xu et al. (2024), we discarded tail categories with fewer than five objects and constructed our training set using 5463 objects from 197 categories in total. The test set is composed of 394 objects, with two left-out objects per category. We used Blender to render the RGB image with its alpha mask from 48 anchor viewpoints at a resolution of 512×512 for both training and testing.

Network Architecture. We leveraged DINO v2 Oquab et al. (2023) as our input image tokenizer, which extracts 768-dimension feature tokens from reference image. The GambaFormer consists of 10 gamba blocks with hidden dimensions 1024, where layer normalization (LN), SSM, depth-wise convolution Shen et al. (2021), and residual connections are employed. The positional embeddings of the GambaFormer consist of 16384 tokens, each with 512 dimensions, corresponding to 16384 3D Gaussians. The Gaussian Decoder is an multi-layer perceptron (MLP) with 10 layers and 64 hidden dimensions, which decodes the output 3D Gaussian of shape (16384, 23) for splatting.

Pre-training. We trained Gamba on 8 NVIDIA A100 (80G) GPUs with batch size 256 for about 3 days. We applied the AdamW optimizer Loshchilov & Hutter (2017) with a learning rate of 1e-4 and a weight decay of 0.05. For each training object, we randomly sample six views among the 48 rendered views to supervise the reconstruction with the input view filtered through CLIP similarity.

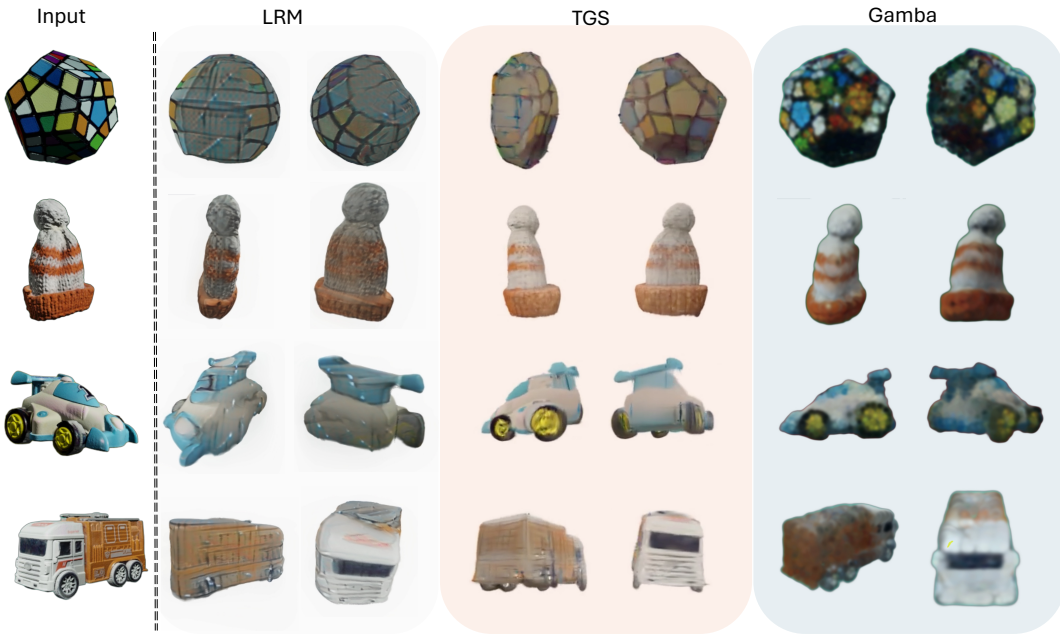

Figure 5: Comparison with feed-forward single-view 3D reconstruction methods.

We set the loss weight $\lambda_{\text{mask}} = 0.01$ and $\lambda_{\text{lpips}} = 0.1$, respectively. Following Tang et al. (2024), we clip the gradient with a maximum norm of 1.0.

Inference. Gamba is an end-to-end feed-forward pipeline that only takes about *8 GB* of GPU memory and less than *1 second* on a single NVIDIA A100 (80G) GPU during inference, facilitating online deployment. Note that Gamba takes an arbitrary RGBA image as input with assumed normalized camera pose, and the output is 16384 numbers of 3D Gaussians for further splatting from any given pose.

5.2 Experimental Protocol

Baseline. We compare our Gamba with previous state-of-the-art single-view reconstruction methods. There are mainly two streams : (1) *Optimization-based methods*, including One-2345 Liu et al. (2023a) and DreamGaussian Tang et al. (2023). One-2345 leverages Zero123 Liu et al. (2023b) and utilizes SparseNeuS Wang et al. (2021; 2023c) to fuse information from noisy multi-view generations; DreamGaussian [*] combines Zero123 Liu et al. (2023b) with 3DGS for efficient 3D generation. (2) *Feed-forward methods*, including Triplane-Gaussian Zou et al. (2023) and LRM Hong et al. (2023b) [†], which are transformer-based reconstruction models with 3DGS and triplane-NeRF as the output 3D representation, respectively. Note that AGG Xu et al. (2024) has not been included for comparison in the current version, as no code has been publicly released.

Evaluation Metrics. In the context of single-view 3D reconstruction, an outstanding 3D model should not only faithfully replicate the reference image but also preserve a consistent relationship with the reference and yield believable outcomes from various viewpoints. As the single-view input can lead to various 3D objects as plausible outputs, we do not evaluate precise matches with the 3D ground truth in the novel view. According to Xu et al. (2024), we evaluated Gamba using the test set of OmniObject3D for a quantitative analysis, employing PSNR and LPIPS metrics to assess reconstruction quality and the CLIP distance defined in the image embedding to reflect the high-level image similarity in multiview renderings.

[*]We only devise the first stage of DreamGaussian without mesh refinement for more fair comparison with the same 3D representation.

[†]LRM is implemented based on the open-source openlrm-mix-base-1.1 model.

Figure 6: Comparison with optimization-based single-view 3D reconstruction methods.

Qualitative Comparisons. Fig. 5 and Fig. 6 demonstrate that Gamba maintains reasonable geometry understanding and plausible texture in most scenarios. On the contrary, the generation results from most baselines, even those that use score distillation sampling (SDS) and more advanced Zero123-XL Liu et al. (2023b), are plagued by multi-view inconsistency, geometric distortion, and texture ambiguity. In particular, when the reference image comes across with an uncanonical view (*e.g.*, the shoes case in the fourth line of Fig. 6, optimization-based methods struggle to produce a proper geometry structure due to the bias in 2D diffusion models. Such an observation leads us to a potential two-stage 3D generation pattern, that is, a feed-forward model,*e.g.*, Gamba, first generates reasonable and consistent geometry and then an optimization-based method, *e.g.*, Consistent3D, further refines the intricate texture and local details with smaller diffusion time-step guidance. See *Appendix* for more details of the above two-stage pattern.

Quantitative Comparisons. Table 1 reveals that Gamba is competitive against other methods across all evaluated metrics, highlighting its exceptional capabilities in both reconstruction quality (PSNR, LPIPS) and 3D consistency (CLIP Distance). Specifically, Gamba matches DreamGaussian, which employs a direct reference view reconstruction supervision with large loss weight, in reconstructing the reference view and substantially outperforms One-2345. Regarding view consistency, evaluated by CLIP Distance, Gamba notably exceeds LRM by a considerable margin. The principal inconsistency found in LRM is attributed to the disparity in edges between concealed and exposed areas, which results in visible seams with blurred textures.

Table 1: **Quantitative results.** We show quantitative results in terms of CLIP Distance (CLIP-D) ↓ / PSNR↑ / LPIPS↓. The above results are shown on the OmniObject3D datasets.

Metrics\Methods	LRM	One-2345	DreamGaussian	**Gamba**
CLIP-D ↓	0.59	0.61	0.41	0.39
PSNR↑	15.80	15.24	23.24	20.19
LPIPS↓	0.49	0.51	0.08	0.15

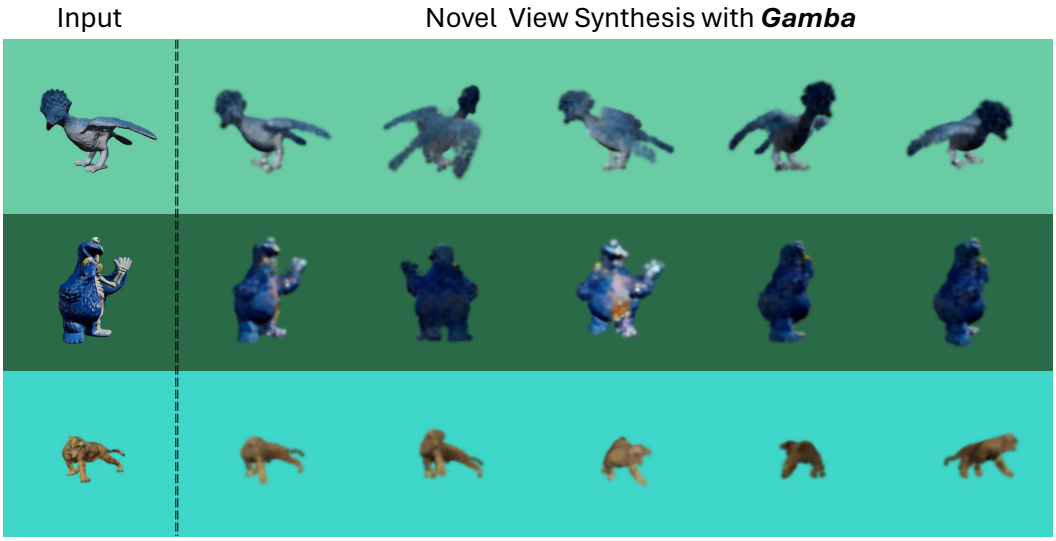

Figure 7: Failure cases of Gamba. Our observation suggests that Gamba struggles to generate 3D assets with extremely intricate texture and out-of-distribution reference inputs.

5.3 Inference Runtime

We showcase the inference runtime required to generate a 3D asset in Table 2, where the timing is recorded using the default hyper-parameters for each method on a single NVIDIA A100 GPU (80G). Remarkably, our Gamba outperforms optimization-based approaches in terms of speed, being several orders of magnitude faster than those optimization-based methods and surpass other feed-forward models as well, thanks to the efficient backbone design.

Table 2: Comparisons against baseline methods regarding inference runtime.

Metrics\Methods	LRM	One-2345	DreamGaussian	**Gamba**
Runtime (s)	5	48	72	0.6

6 Ablation and Discussion

Q1: ***What impacts performance of Gamba in terms of component-wise contributions?*** We discarded each core component of Gamba to validate its component-wise effectiveness. The results are described in Table 3 by comparing their predictions with **3D ground truth** on the test set.

A1: In an end-to-end, multi-component pipeline, we observed that the exclusion of any component from Gamba resulted in a significant degradation in performance. In particular, we first replace the proposed GambaFormer block with the traditional transformer-based structure utilizing adaptive layer norm (adaLN) and cross-attention for condition control. Such "w/o Mamba Block" is limited in the number of tokens (3D Gassians), thus delivering geometric irregularities and excessive texture details with extremely lower PSNR and SSIM. In addition, the "w/o Robust Training" appears with more floaters and worse geometry, as it is easier to get stuck in the local minimum and dimensional collapse, resulting in 3DGS parameters with little standard deviation. In contrast, integrated with the mamba block and robust training, Gamba showcases the best rendering outcomes in the novel view. More qualitative comparisons will be included in *Appendix*.

Q2: ***How about the failure case of Gamba?*** As shown in Fig. 7, we manually selected some failure case from the Gamba generation results.

Table 3: Ablation Studies. We validate the effectiveness of our proposed structure and training design.

Models\Metrics	PSNR ↑	SSIM ↑	LPIPS ↓
w/o Mamba Block	11.28	0.60	0.22
w/o Robust Training	15.44	0.69	0.21
Gamba	19.20	0.82	0.19

A2: To be honest, our Gamba does not always yield plausible results. **First**, Gamba struggles with generating sharp textures for occluded areas, particularly when dealing with complex textures. We hypothesize that this issue arises from two main factors: (1) As highlighted in LRM Hong et al. (2023b), the task of single-view 3D reconstruction is inherently probabilistic, suggesting multiple potential solutions for unseen areas. Nonetheless, our model operates on a deterministic basis, likely leading to the production of averaged representations for these unseen regions. (2) Gamba simultaneously predicts and supervises both geometric (e.g., position) and appearance information (e.g., spherical harmonics), which hinders its ability to accurately model intricate textures and scene illumination. **Second**, given that Gamba has only been pre-trained on the OmniObject3D dataset Wu et al. (2023)—which is significantly smaller than the widely-adopted Objaverse dataset Deitke et al. (2023)—it struggles to accurately reconstruct 'unseen' 3D assets that exhibit a large domain disparity from the scanned objects in OmniObject3D. We plan to release an enhanced version of Gamba, pre-trained on the filtered subset of Objaverse-XL in the near future.

7 Conclusion

In this technique report, we present Gamba, an end-to-end, amortized 3D reconstruction model from single-view image. Our proposed Gamba, different from previous methods reliant on SDS and NeRF, marries 3D Gaussian splatting and Mamba to address the challenges of high memory requirements and heavy rendering process. Our key insight is the relationship between the 3DGS generation process and the sequential mechanism of Mamba. Additionally, Gamba integrates several techniques for training stability. Through extensive qualitative comparisons and quantitative evaluations, we show that our Gamba is promising and competitive with several orders of magnitude speedup in single-view 3D reconstruction.

References

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5855–5864, 2021.

David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. *arXiv preprint arXiv:2312.12337*, 2023.

Guikun Chen and Wenguan Wang. A survey on 3d gaussian splatting. *arXiv preprint arXiv:2401.03890*, 2024.

François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6569–6578, 2019.

William Glasser. *Control theory*. Harper and Row New York, 1985.

Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *Advances in Neural Information Processing Systems*, 35:14715–14728, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023a.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023b.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021a.

Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021b.

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022a.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Ré. How to train your hippo: State space models with generalized orthogonal basis projections. *arXiv preprint arXiv:2206.12037*, 2022b.

Faiza Gul, Wan Rahiman, and Syed Sahal Nazli Alhady. A comprehensive study for robot navigation techniques. *Cogent Engineering*, 6(1):1632046, 2019.

Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. `https://github.com/threestudio-project/threestudio`, 2023.

Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for robust text-to-3d generation. *arXiv preprint arXiv:2303.15413*, 2023a.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023b.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023.

Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024a.

Dingkang Liang, Xin Zhou, Xinyu Wang, Xingkui Zhu, Wei Xu, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. *arXiv preprint arXiv:2402.10739*, 2024b.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 300–309, 2023.

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. *arXiv preprint arXiv:2306.16928*, 2023a.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309, 2023b.

Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jun Ma, Feifei Li, and Bo Wang. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*, 2024.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. Long range language modeling via gated state spaces. *arXiv preprint arXiv:2206.13947*, 2022.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*, pp. 7220–7229. PMLR, 2020.

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

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *arXiv preprint arXiv:2306.17843*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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

Jiacheng Ruan and Suncheng Xiang. Vm-unet: Vision mamba unet for medical image segmentation. *arXiv preprint arXiv:2402.02491*, 2024.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

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

Tianyu Sun, Guodong Zhang, Wenming Yang, Jing-Hao Xue, and Guijin Wang. Trosd: A new rgb-d dataset for transparent and reflective object segmentation in practice. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. *arXiv preprint arXiv:2312.13150*, 2023.

Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653*, 2023.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024.

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

Jue Wang, Wentao Zhu, Pichao Wang, Xiang Yu, Linda Liu, Mohamed Omar, and Raffay Hamid. Selective structured state-spaces for long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6387–6397, 2023a.

Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.

Peng Wang, Hao Tan, Sai Bi, Yinghao Xu, Fujun Luan, Kalyan Sunkavalli, Wenping Wang, Zexiang Xu, and Kai Zhang. Pf-lrm: Pose-free large reconstruction model for joint pose and shape prediction. *arXiv preprint arXiv:2311.12024*, 2023b.

Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3295–3306, 2023c.

Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3d editing with gaussian splatting. *arXiv preprint arXiv:2403.11868*, 2024.

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. *arXiv preprint arXiv:2305.16213*, 2023d.

Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092*, 2023.

Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 803–814, 2023.

Zike Wu, Pan Zhou, Xuanyu Yi, Xiaoding Yuan, and Hanwang Zhang. Consistent3d: Towards consistent high-fidelity text-to-3d generation with deterministic sampling prior. *arXiv preprint arXiv:2401.09050*, 2024.

Dejia Xu, Ye Yuan, Morteza Mardani, Sifei Liu, Jiaming Song, Zhangyang Wang, and Arash Vahdat. Agg: Amortized generative 3d gaussians for single image to 3d. *arXiv preprint arXiv:2401.04099*, 2024.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. Dmv3d: Denoising multi-view diffusion using 3d large reconstruction model. *arXiv preprint arXiv:2311.09217*, 2023.

Xuanyu Yi, Kaihua Tang, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. Identifying hard noise in long-tailed sample distribution. In *European Conference on Computer Vision*, pp. 739–756. Springer, 2022.

Xuanyu Yi, Jiajun Deng, Qianru Sun, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. Invariant training 2d-3d joint hard samples for few-shot point cloud recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14463–14474, 2023.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9150–9161, 2023.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Tao Zhang, Xiangtai Li, Haobo Yuan, Shunping Ji, and Shuicheng Yan. Point could mamba: Point cloud learning via state space model, 2024.

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024.

Zi-Xin Zou, Zhipeng Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Yan-Pei Cao, and Song-Hai Zhang. Triplane meets gaussian splatting: Fast and generalizable single-view 3d reconstruction with transformers. *arXiv preprint arXiv:2312.09147*, 2023.

Matthias Zwicker, Hanspeter Pfister, Jeroen Van Baar, and Markus Gross. Ewa splatting. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):223–238, 2002.