DreamBeast: Distilling 3D Fantastical Animals with Part-Aware Knowledge Transfer

Runjia Li[1] Junlin Han[1] Luke Melas-Kyriazi[1] Chunyi Sun[2] Zhaochong An[3]
Zhongrui Gui[1] Shuyang Sun[1] Philip Torr[1] Tomas Jakab[1]
[1]University of Oxford [2]Australian National University [3]University of Copenhagen
dreambeast3d.github.io

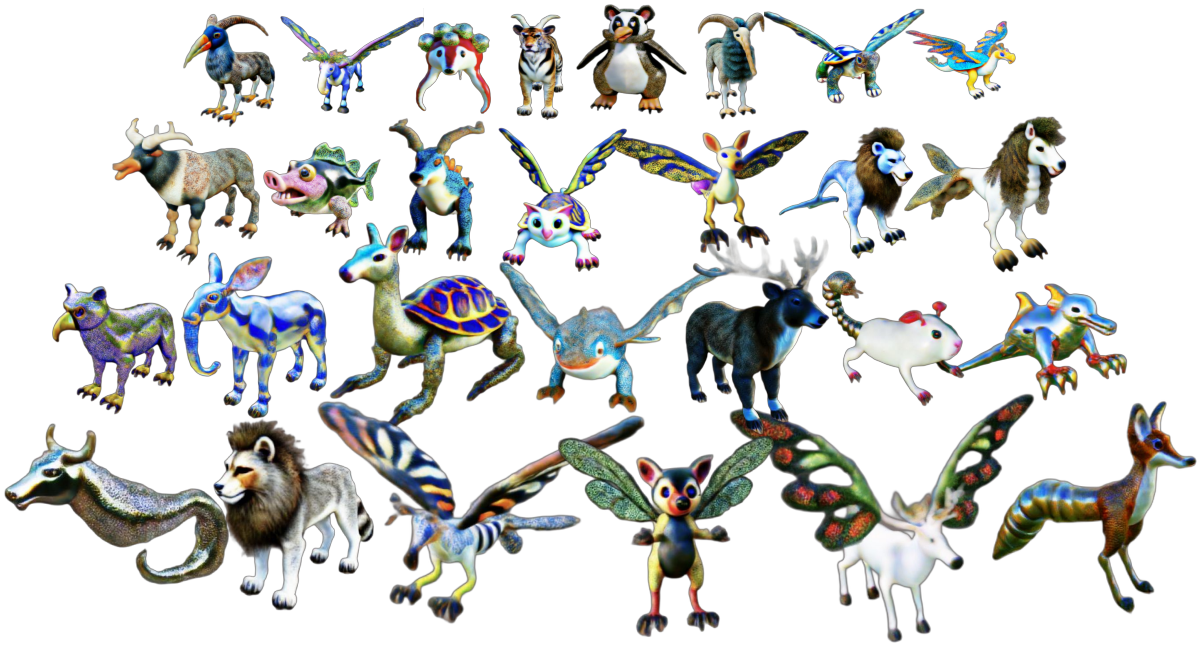

Figure 1. **Generated fantastic 3D beasts composed of diverse animal parts.** Our method enables part-level generation, resulting in 3D creatures with unique combinations of heads, limbs, wings, tails, and bodies.

Abstract

We present DreamBeast, a novel method based on score distillation sampling (SDS) for generating fantastical 3D animal assets composed of distinct parts. Existing SDS methods often struggle with this generation task due to a limited understanding of part-level semantics in text-to-image diffusion models. While recent diffusion models, such as Stable Diffusion 3, demonstrate a better part-level understanding, they are prohibitively slow and exhibit other common problems associated with single-view diffusion models. DreamBeast overcomes this limitation through a novel part-aware knowledge transfer mechanism. For each generated asset, we efficiently extract part-level knowledge from the Stable Diffusion 3 model into a 3D Part-Affinity implicit representation. This enables us to instantly generate Part-Affinity maps from arbitrary camera views, which we then use to modulate the guidance of a multi-view diffusion model during SDS to create 3D assets of fantastical animals. DreamBeast significantly enhances the quality of generated 3D creatures with user-specified part compositions while reducing computational overhead, as demonstrated by extensive quantitative and qualitative evaluations.

1. Introduction

Imagine a creature taking flight, its dragon wings catching the sunlight. Picture its majestic lion's head surveying the landscape, while a sinuous serpent's tail trails behind.

What if "Fantastic Beasts and Where to Find Them" was not just a magical story, but we could actually build them in a digital 3D world? Current methods for generating 3D objects [1, 3, 15, 18, 38, 48, 53] struggle with generating complex, artistic, or fantastical shapes and textures, which are not represented in existing datasets. For example, they are unable to produce Griffin-like animals composed of parts from multiple species. More generally, they struggle with producing objects composed of multiple diverse parts.

One of the most promising current approaches to open-world 3D asset generation consists of lifting 2D guidance into 3D. Methods such as DreamFusion [32] and SJC [44] demonstrate how pre-trained 2D diffusion models [12, 34, 35, 37] can guide the generation of 3D objects through score distillation sampling (SDS). Specifically, these methods produce 3D objects from textual descriptions by utilizing the priors encoded in 2D diffusion models, which act as approximate log gradients of the density of distribution of 2D images conditioned on text.

The lifting methods, however, fall short of providing part-level controllability for part-specific textual descriptions. The reason for this is twofold. First, there have not been any 2D diffusion models capable of sufficiently strong part-level understanding. Second, in part due to the first reason, there have been no methods proposed in the literature for part-aware lifting-based (SDS) text-to-3D generation from part-specific textual prompts.

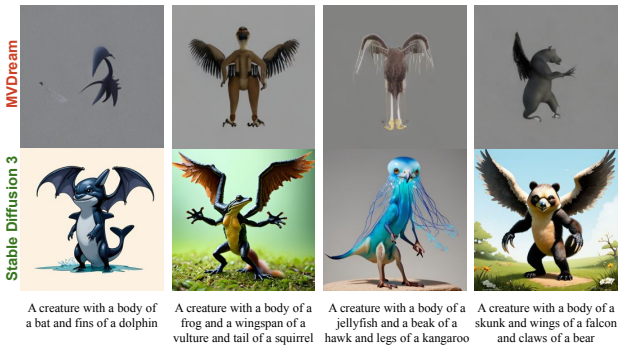

Figure 2. **Comparison of diffusion models on part-level prompt understanding in 2D generation.** Although MVDream can grasp the overall semantic understanding of the described animals, the generated images often feature deformed animals and fail to accurately capture specific part-based descriptions, unlike SD3.

Diffusion models are improving rapidly, and the recent release of Stable Diffusion 3 (SD3) [12] has led us to reconsider the subject of part-level generation. The starting point for our paper is the observation that SD3 can capture part-level correspondences significantly better than prior models (as shown quantitatively in Table 1 and qualitatively in Figure 2). This new capability allows for the generation of complex part-aware entities through part-specific text descriptions. However, such fine-grained understanding capabilities are not yet available for 3D generation.

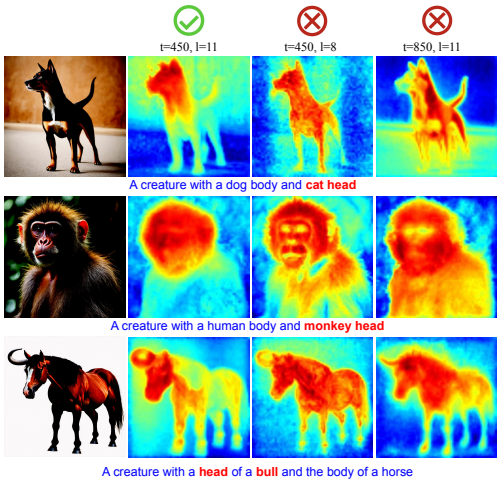

Figure 3. **Failing to generate part-aware content even with part understanding in SD3.** Despite its understanding of part correspondences, as evidenced by the cross-attention maps at certain timesteps t and layers l, SD3 may still fail to generate part-aware images. This is illustrated in above examples where specific animal parts are absent, highlighted in red. Our method capitalizes on the observation that only particular timesteps t and layers l exhibit part-awareness.

As SDS is potentially capable of lifting any entities from 2D to 3D, a straightforward approach might combine SD3 with SDS. However, as shown in Figure 3, we observe that SD3 occasionally struggles to generate animals according to prompts that specify particular animal body parts, even though it understands where those parts should be in the cross attention maps. This issue arises because part-correspondence understanding is only reliable at certain timesteps and transformer layers, making the SDS process less robust to prompts that focus on specific parts. Additionally, naively using SD3 within SDS is leading to multiple issues. SD3's use of the rectified flow match Euler discrete scheduler results in deformed outputs with standard timestep sampling used for other diffusion models, as seen in Figure 4 and 9. Other issues associated with SDS such as multi-face Janus problem and content drift [37] are also present. Furthermore, generating 3D assets with SD3 takes 7 hours, which can be prohibitive in many applications.

To overcome the aforementioned challenges, we introduce DreamBeast, a novel part-aware knowledge transfer module designed to efficiently distill part-level understanding from powerful single-view diffusion models, such as SD3, and inject it into 3D generation with SDS.

DreamBeast first performs SDS over several steps to partially optimize the NeRF, producing a coarse yet view-consistent layout of the animal's shape. We then render the

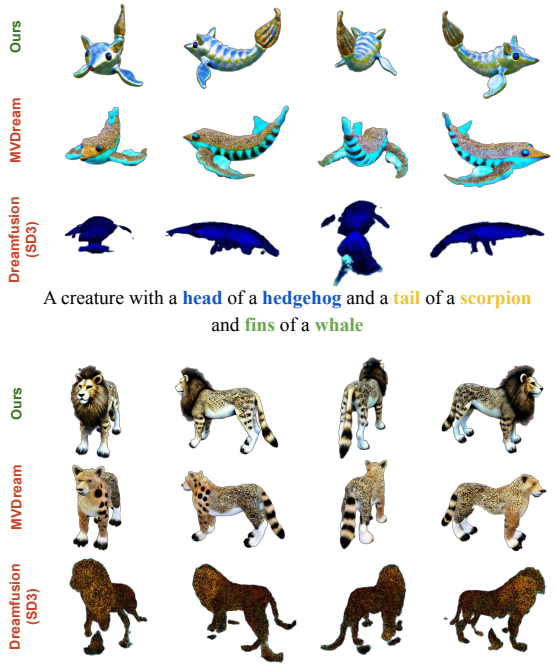

Figure 4. **MVDream and SD3 have difficulty generating part-aware 3D animals.** While SD3 [12] can understand part correspondence in images and text, it struggles to generate 3D assets using SDS due to the issues we discussed in our paper. MVDream [37] falls short because it was fine-tuned on Objaverse [9], which lacks part-level information in the dataset.

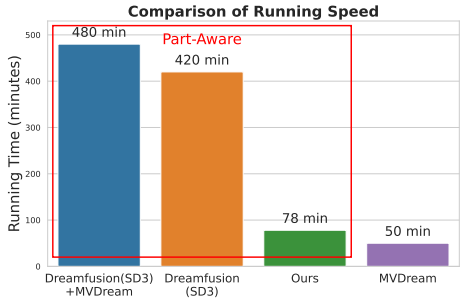

Figure 5. **Running speed comparison.** While Dreamfusion (SD3) combined with MVDream and standalone Dreamfusion (SD3) take 480 and 420 minutes respectively, our method significantly reduces the runtime to 78 minutes. This reduction is achieved without sacrificing part-awareness making our method both faster and more effective in part-aware 3D generation.

NeRF from a limited number of camera viewpoints. We use these renderings as the denoising condition for SD3 and perform several denoising steps during which we extract cross-attention maps of part-specific tokens from part-aware layers. We average these cross-attention maps for each of the camera viewpoints and obtain part affinity maps.

Next, we train a 3D Part-Affinity representation based on NeRF from the extracted part affinity maps, which allows us to interpolate part affinity maps from any camera viewpoint almost instantaneously. Subsequently, during SDS, we render both the 3D asset and the learned Part-Affinity NeRF of DreamBeast from the same camera perspective and modulate the cross and self-attention mechanisms of the guidance model using the rendered part affinity map. This modulation causes regions with high part affinity to have higher responses to part-specific prompts. Consequently, our approach (DreamBeast) not only promotes more reliable part-aware SDS but also significantly reduces the computational cost from 7 hours to 78 minutes and cuts GPU memory usage by 24GB, compared to the combination of SD3 with SDS as demonstrated in Figure 5. Quantitative and qualitative evaluations of DreamBeast demonstrate its exceptional ability to generate part-aware, imaginative 3D animals.

In summary, our **main contributions** are as follows:

1. We are the first to consider the problem of part-level text to 3D generation in an open-world setting.
2. We propose DreamBeast, a novel knowledge transfer mechanism that efficiently transfers part-level knowledge of a 2D diffusion model into the 3D generation process.
3. We significantly improve the quality and decrease the computational cost of creating part-aware 3D animal assets by integrating DreamBeast within the SDS optimization process.
4. We demonstrate through quantitative evaluations and a human study that our method consistently outperforms baseline methods.

2. Related Work

Lifting 2D Diffusion Models for 3D Animal Generation. Due to the limited generalizability of current 3D generative models, efforts have been made to adapt 2D diffusion priors or single image/video [24, 49] for 3D assets such as animals. The distilling diffusion prior approach primarily employs score distillation sampling (SDS) [32], where 2D diffusion priors serve as score functions that guide the optimization of 3D structures. Similarly, SJC [44] utilized publicly accessible diffusion models for their method. Subsequent studies have aimed at refining 3D representations, enhancing loss design, or implementing multi-stage optimizations [6, 7, 19, 25, 41, 46, 51, 52]. Some methods [20, 27, 33, 39] leverage diffusion guidance to optimize 3D models based on a single image. Another set of methods uses diffusion guidance to learn the layout for compositional generation [5, 11, 31]. Notably, MVDream [37] proposed a multi-view consistent diffusion model for SDS guidance, significantly improving issues with multi-face Janus and content drift.

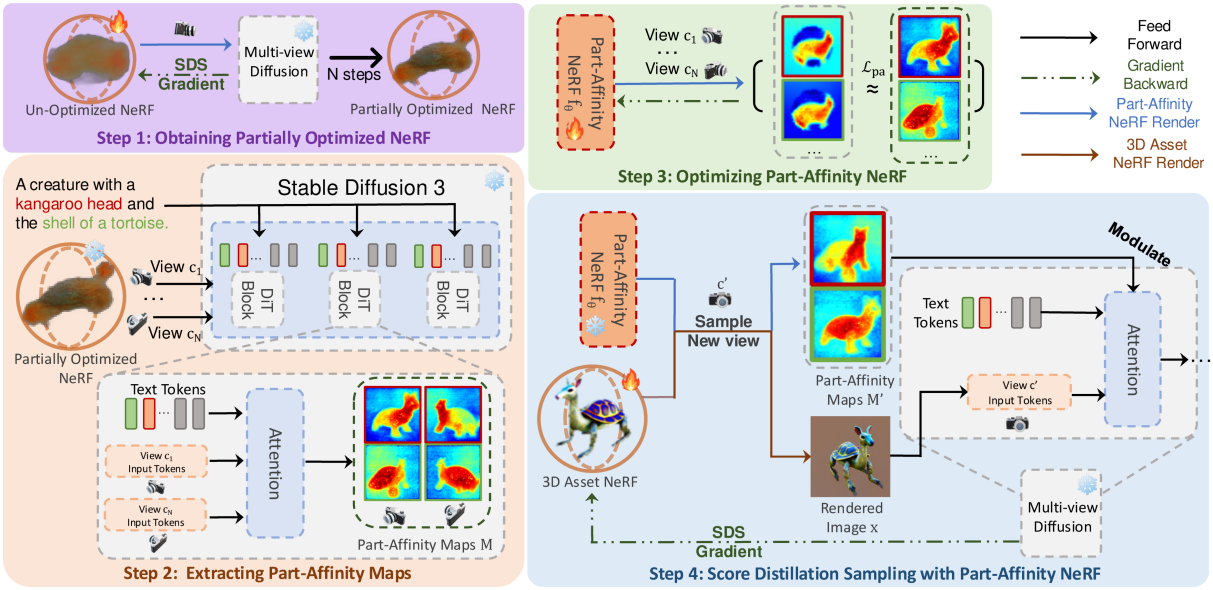

Figure 6. **DreamBeast pipeline.** DreamBeast is composed of 4 steps: (1)Partially optimize a NeRF under standard SDS. (2) Multiple rendered views of the partially optimized NeRF are input into SD3 along with the text prompt to construct Part-Affinity maps based on cross-attention of part tokens in SD3. (3) A Part-Affinity NeRF is trained using these extracted maps. (4) Both the trainable 3D asset NeRF and frozen Part-Affinity NeRF are rendered from the same camera pose. The rendered Part-Affinity map then modulates cross and self-attention in MVDream for SDS, ultimately generating a part-aware 3D animal. The symbol ❄ denotes a frozen model, while 🔥 indicates a model that is trainable.

Layout Guided 3D Generation. Earlier research [29] has explored the application of compositional neural radiance fields within an adversarial learning framework to achieve 3D-aware image generation. A pioneering study [13] utilized a mesh database to find and combine parts to create new objects. Subsequent research incorporated probabilistic models for part suggestion [21], semantic attributes [2], and fabrication [36]. Some studies [42] employed neural radiance fields to represent various 3D elements and render these into a unified 3D model. Recent advancements [31], guided by pre-trained diffusion models, have enabled the generation of compositional 3D scenes using user-provided 3D bounding boxes and text prompts. Concurrently, other works have used large language models (LLMs) to generate 3D layouts from text prompts as an alternative to human annotations [14, 43, 45], or combined layout learning during the optimization process [5, 11]. While these approaches can produce 3D scenes through composition, they all rely heavily on scene graphs or descriptions of object-to-object relationships for object-to-scene generation. This dependency makes it impossible when it comes to composing part-to-object generational tasks. Unlike object-to-scene generation, which requires an understanding of the relationships between distinct objects, part-to-object generation necessitates a more fine-grained comprehension of how individual parts combine to form a coherent whole.

Diffusion with Cross-Attention Control. Since most current diffusion models are transformer-based and incorporate text information through cross-attention layers, providing spatial awareness naturally aligns with cross-attention control. Several studies [4, 10, 17, 23, 30] explore various methods to enhance cross-attention scores between regions and their corresponding descriptions in the prompts. In contrast, others [16] propose applying a binary mask to eliminate attention between regions and non-matching region descriptions. To the best of our knowledge, cross-attention control is predominantly applied in 2D generation models and is seldom utilized in 3D SDS settings as in our paper.

3. Method

To efficiently transfer part-level knowledge of a 2D diffusion model into the 3D generation process, we introduce DreamBeast as a novel geometry-consistent mechanism designed for this purpose. In Section 3.1, we revisit the classic Score Distillation Sampling (SDS) formulation and discuss the issues that arise when applying SDS directly with SD3. Subsequent sections detail our method: Section 3.2 describes the extraction of 2D part affinity maps from SD3, Section 3.3 details the construction of our Part-Affinity NeRF using the part affinity maps, and Section 3.4 presents the integration of the Part-Affinity NeRF within SDS to generate part-aware 3D assets.

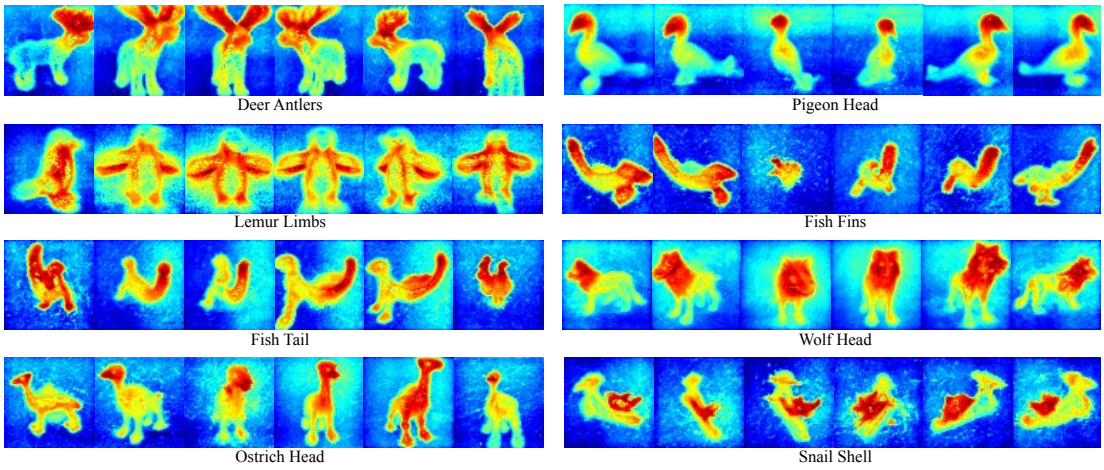

Figure 7. **Learned part affinity map rendered from unseen camera poses.** Heatmaps displaying the learned 3D part affinity representation rendered from unseen camera poses for different part-specific descriptions of distinct animals. Warmer colors indicate stronger affinities, highlighting our implicit 3D neural representation's capability to differentiate and localize specific anatomical features.

3.1. Preliminaries

Before delving into our method, we briefly review the concepts commonly employed in the 3D lifting generation techniques and further discuss our motivation.

Score Distillation Sampling. As introduced by [32], Score Distillation Sampling (SDS) uses a pre-trained 2D diffusion model with fixed parameters ϕ to guide the generation of 3D models with the vast amount of 2D image prior knowledge. Let θ denote the 3D representation such as NeRF [28] or Gaussian Splatting [22] and $g(\cdot)$ be the differentiable rendering function, which renders the 3D model to an image $x = g(\theta)$. During the guidance process with y as the text condition, we first sample a random timestep $t \sim \{0, ..., T\}$ and a random noise $\epsilon \sim \mathcal{N}(0, I)$. We then add the noise to the rendered image and we get $x_t = \sqrt{\overline{\alpha_t}}x + \sqrt{1 - \overline{\alpha_t}}\epsilon$. The SDS gradient then is defined as follows:

$$\nabla_\theta \mathcal{L}_{SDS}(x = g(\theta)) = \mathbb{E}_{t,\epsilon}\left[w(t)(\hat{\epsilon}(x_t; y, t, \phi) - \epsilon)\frac{\partial x}{\partial \theta}\right]$$

where $\overline{\alpha_t}$ and $w(t)$ are weighting functions that depend on the timestep t, and $\hat{\epsilon}$ is the predicted noise by the pre-trained diffusion model.

Why not Stable Diffusion 3 as the Guidance Directly? A straightforward way to leverage part-level knowledge from SD3 is by performing SDS with SD3's guidance. However, we argue that this approach is ineffective for several reasons: (1) Even when SD3 manages to generate images with part-level specifications, it often fails because part-level understanding is only exhibited at specific transformer layers and timesteps, as shown in Figure 3. During the denoising forward pass through SD3, information about the animals and their parts can become mixed, leading to a loss of part-level control. This makes SDS unstable and unpredictable at the part level. (2) SD3 cannot provide view-consistent guidance, which leads to issues like multi-face Janus problem and content drift problems [37]. (3) SD3 uses a rectified flow match Euler discrete scheduler, which differs from previous diffusion methods. We observe that timestep sampling strategies from earlier methods do not yield satisfactory results with SD3. Additionally, hyperparameters such as the guidance scale and 3D shape loss scales require extensive empirical tuning. (4) The forward process is computationally expensive, requiring 48GB of GPU memory and 7 hours of training time on an NVIDIA A40 to generate just one 3D asset. In contrast, DreamBeast offers more stable and robust part-level specification during SDS and requires only 78 minutes to complete. This is slightly longer than SD2.1 or MVDream, which takes 50 minutes on an A40 GPU (see Figure 5 for a comparison of running speeds).

Why not Stable Diffusion 3 + MVDream as the Guidance? This approach could potentially solve the multi-face Janus problem; however, the remaining issues with SD3 still persist. Moreover, this combination requires 58GB of GPU memory and takes 8 hours to generate a single 3D asset through SDS, rendering it highly inefficient.

3.2. Part Affinity Map Extraction

Before extracting the part affinity map from SD3, we first perform SDS for several steps to partially optimize the NeRF. We then render this partially optimized NeRF from various camera angles to obtain view-consistent, coarse animal-shape layouts. These view-consistent layouts serve as conditions for SD3, where the rendered animal shape is

mixed with noise as input for denoising, which helps keep the extracted part affinity maps view-consistent as well.

We choose Stable Diffusion 3 (SD3) as our source of 2D part-level knowledge for two key reasons: (1) Unlike perception-driven part-level segmentation frameworks [8, 40, 47, 50] that rely on well-defined input images, we found that SD3 can operate effectively on noisy images generated from partially optimized NeRFs.

(2) Among many open-source models we examined, SD3 is the only one that demonstrates part-level understanding in its cross-attention maps.

To obtain the part affinity map, we conduct a denoising process for the timesteps between t_s and t_e. Specifically, for a rendered image x from a partially optimized NeRF under camera pose c, we introduce noise to x using a weighting factor α_{t_s} to produce x_{t_s}. Subsequently, we perform denoising in the latent space for each timestep up to t_e. At each timestep t, and for each transformer layer l in SD3, we compute an attention map $A_{t,l,c} \in \mathbb{R}^{HW \times n}$, where n denotes the number of tokens in the text prompt $y \in \mathbb{R}^n$, and HW represents the spatial resolution of the feature maps. From this attention map, we extract a spatial correspondence map $M_{t,l,i,c} \in \mathbb{R}^{HW}$ for each token y_i, which corresponds to a slice of $A_{t,l,c}$ associated with the token y_i. Let I_p represent the set of token indices for the part-level description in y, for example, "kangaroo head" in "a creature with a kangaroo head and a tortoise shell" corresponds to a set of token indices $I_p = \{4, 5\}$. We can then derive the part affinity map $M_{p,c} \in \mathbb{R}^{HW}$ for this camera pose using the following equation:

$$M_{p,c} = \frac{1}{(t_s - t_e) \cdot L \cdot |I_p|} \sum_{t=t_s}^{t_e} \sum_{l=0}^{L} \sum_{i \in I_p} M_{t,l,i,c}$$

where L is the number of transformer layers.

3.3. Part-Affinity NeRF

The part affinity map $M_{p,c}$ applies only to a specific camera pose c. This poses a problem, as the camera poses are sampled from a continuous distribution during SDS, implying an infinite number of potential camera poses.

A naive approach would be to generate the part affinity map each time an image is rendered from the 3D asset. However, this approach significantly increases computational demands. The forward pass in SD3, combined with the need to obtain cross-attention maps for every layer and timestep, requires substantially more computation. As a result, generating a single 3D asset can take up to 58 hours, which is even longer than using SD3 solely as a guidance mechanism for SDS.

Therefore, as part of DreamBeast, we introduce the Part affinity NeRF that learns from part affinity maps of discrete camera poses and is capable of interpolating these maps for continuous camera poses.

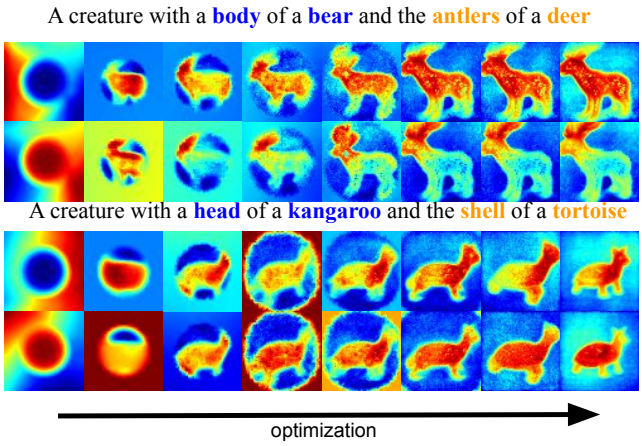

Figure 8. **Part-Affnity NeRF learning progress visualization.** We show the evolution of renderings of the Part-Affinity NeRF for each corresponding part (e.g., body of a bear in the first row) throughout the learning, demonstrating that the Part-affinity NeRF quickly converges.

Mathematically, let $\{\{M_{p,c} \mid p \in \mathcal{P}\} \mid c \in \mathcal{C}\}$ denote all part affinity maps obtained for all parts $\mathcal{P}$ under a set of camera poses $\mathcal{C}$. The optimization goal is to fit the implicit representation $f_\theta(M \mid p, c)$ parameterized by a neural network with parameters θ :

$$\min_\theta \mathcal{L}_{pa} = \min_\theta \sum_{c \in \mathcal{C}} \sum_{p \in \mathcal{P}} ||f_\theta(M \mid p, c) - M_{p,c}||^2$$

In our design, f_θ is an MLP-based neural radiance field. The continuous nature of MLPs imposes a form of smoothness and continuity in the learned representation, enabling DreamBeast to produce part affinity maps from any camera pose. In conclusion, DreamBeast offers several key benefits: (1) it provides a 3D consistent representation of part affinity maps learned from discrete camera views, allowing for the interpolation of part affinity maps under any camera pose, which makes it efficient to use with the SDS process; (2) it significantly reduces the computation cost of using SD3 at every single step to the more economical learning and rendering cost of DreamBeast, reducing the training time from 58 hours to 78 minutes per 3D asset.

3.4. SDS with Attention Modulation

After rendering an image $x(c')$ from the 3D asset under a specific camera pose c' during SDS, we also generate the rendered affinity maps for each of the parts $\{f_\theta(M \mid p, c') | p \in \mathcal{P}\}$ using the optimized Part-Affinity NeRF under the same camera pose. These rendered part affinity maps are utilized to modulate the cross-attention and self-attention matrices in the 2D diffusion guidance model for SDS. Specifically, we modulate the cross-attention score maps $\mathcal{S}_{cross} \in \mathbb{R}^{hw \times n}$ and the self-attention score maps

Algorithm 1 Layer-wise Attention Modulation with DreamBeast

Require: Cross attention and self attention of the 2D diffusion guidance model $\mathcal{S}_{cross}, \mathcal{S}_{self}$, rendered part affinity map $\{f_\theta(M \mid p, c') | p \in \mathcal{P}\}$, part token indices $\{I_p \mid p \in \mathcal{P}\}$, enhancement factor α_{cross}, α_{self}

for p in $\mathcal{P}$ **do** ▷ Cross Attention
  $M'_{p,c'} \leftarrow f_\theta(M \mid p, c')$
  $\mathcal{S}_{cross}[:, I_p] \leftarrow \mathcal{S}_{cross}[:, I_p] + \alpha_{cross} \log M'_{p,c'}$
  $A_{cross} \leftarrow \text{Softmax}(\mathcal{S}_{cross})$
end for
for p in $\mathcal{P}$ **do** ▷ Self Attention
  $M'_{p,c} \leftarrow f_\theta(M \mid p, c')$
  $K_{p,c'} \leftarrow (M'_{p,c'})^T M'_{p,c'}$ ▷ Symmetry
  $\mathcal{S}_{self} \leftarrow \mathcal{S}_{self} + \alpha_{self} \log K_{p,c'}$
  $A_{self} \leftarrow \text{Softmax}(\mathcal{S}_{self})$
end for

2D Diffusion Model	2 Parts	3 Parts
MVDream [37]	0.242	0.108
Stable Diffusion 2.1 [34]	0.187	0.022
Stable Diffusion XL	0.297	0.032
DeepFloyd [35]	0.429	0.097
Stable Diffusion 3 [12]	**0.826**	**0.537**

Table 1. **2D part-aware generation success rate.** A user study involving five participants found that SD3 has a significantly higher success rate than other popular diffusion models in generating part-aware images based on part-level prompts (describing 2 or 3 animal parts in a single prompt). This suggests that SD3 has a superior ability to understand and generate images at the part level.

$\mathcal{S}_{self} \in \mathbb{R}^{hw \times hw}$ (before the softmax operation), where hw represents the feature spatial resolution and n denotes the number of tokens, in the 2D diffusion guidance models at each denoising step $\hat{\epsilon}(x_t; y, t, \phi)$. The detailed procedure is outlined in Algorithm 1. The cross-attention modulation ensures that regions corresponding to a specific part are guided by their corresponding part-specific token. The self-attention modulation increases influence within intra-part regions and reduces influence among inter-part regions.

4. Experiments

4.1. Implementation Details

We consider the cross-attention map between $t_s = 450$ and $t_e = 100$ at the eleventh layer ($l = 11$) in SD3, where we found the most significant part-level understanding.

Our Part-Affinity NeRF is a small MLP with one hidden layer comprising 16 neurons. The output dimension is equivalent to the number of parts described in the prompt, with each dimension representing a different part of the fantastical animal. The part affinity map is rendered like NeRF [28], using 128 samples per ray and a resolution of 64. We optimize Part-Affinity NeRF for 2000 steps.

We choose MVDream [37] as our diffusion guidance model due to its multiview consistency and computational efficiency for SDS. We set $\alpha_{cross} = 0.8$ and $\alpha_{self} = 0.9$ to modulate the influence of cross and self-attention maps within MVDream. For the SDS optimization, we adhere to the default hyper-parameters specified in MVDream for other settings. The training procedure involves several steps to create a detailed 3D asset of an animal. Initially, the asset is trained (guided by MVDream) for 1000 steps to obtain the partially optimized NeRF. Following this, the Part-Affinity NeRF is optimized over an additional 2000 steps. Finally, the training continues for another 4000 steps, using part-level guidance from the rendered part affinity maps, to achieve the final part-aware 3D model.

Method	CLIP Score↑		
	B/32	B/16	L/14
DreamFusion(SD2.1) [32]	$0.271 \pm 3.0e^{-4}$	$0.274 \pm 2.2e^{-4}$	$0.226 \pm 3.6e^{-4}$
DreamFusion(SD3) [32]	$0.271 \pm 5.8e^{-4}$	$0.275 \pm 5.5e^{-4}$	$0.229 \pm 7.7e^{-4}$
MVDream [37]	$0.275 \pm 7.8e^{-4}$	$0.282 \pm 4.1e^{-4}$	$0.230 \pm 7.9e^{-4}$
GeoDream [26]	$0.244 \pm 1.2e^{-4}$	$0.252 \pm 1.5e^{-4}$	$0.202 \pm 1.8e^{-4}$
OpenLRM [18]	$0.265 \pm 2.1e^{-4}$	$0.285 \pm 2.4e^{-4}$	$0.223 \pm 6.2e^{-4}$
VFusion3D [15]	$0.268 \pm 1.9e^{-4}$	$0.281 \pm 2.1e^{-4}$	$0.225 \pm 6.2e^{-4}$
DreamBeast (Ours)	$\mathbf{0.285} \pm 2.6e^{-4}$	$\mathbf{0.289} \pm 3.5e^{-4}$	$\mathbf{0.245} \pm 4.6e^{-4}$

Table 2. **Performance comparison of different methods.** Our method shows the best CLIP scores among all.

View Number	CLIP Score↑		
	B/32	B/16	L/14
8	$0.274 \pm 5.0e^{-4}$	$0.281 \pm 4.6e^{-4}$	$0.231 \pm 5.9e^{-4}$
16	$0.277 \pm 4.8e^{-4}$	$0.281 \pm 1.9e^{-4}$	$0.232 \pm 5.8e^{-4}$
32	$0.277 \pm 4.5e^{-4}$	$0.283 \pm 4.3e^{-4}$	$0.235 \pm 5.7e^{-4}$
64	$0.284 \pm 4.3e^{-4}$	$0.287 \pm 4.0e^{-4}$	$0.240 \pm 5.5e^{-4}$
76	$0.285 \pm 2.6e^{-4}$	$0.289 \pm 3.5e^{-4}$	$0.245 \pm 4.6e^{-4}$

Table 3. **Ablation study on the number of views for extracted part-affinity maps.** Increasing the number of views results in a stronger part-affinity NeRF.

4.2. Evaluation Benchmarks

We use GPT-4o-mini to randomly generate 30 prompts under a template "a creature with a [animal 1][part 1], [animal 2][part 2], and [animal 3][part 3]". We use CLIP text similarity and ranking-based user study to evaluate how well the generated 3D assets match their descriptions.

4.3. Main Results

CLIP Similarity Experiment. We compare DreamBeast with other distillation-based methods [26, 32, 37], and popular feedforward 3D generation methods [15, 18],

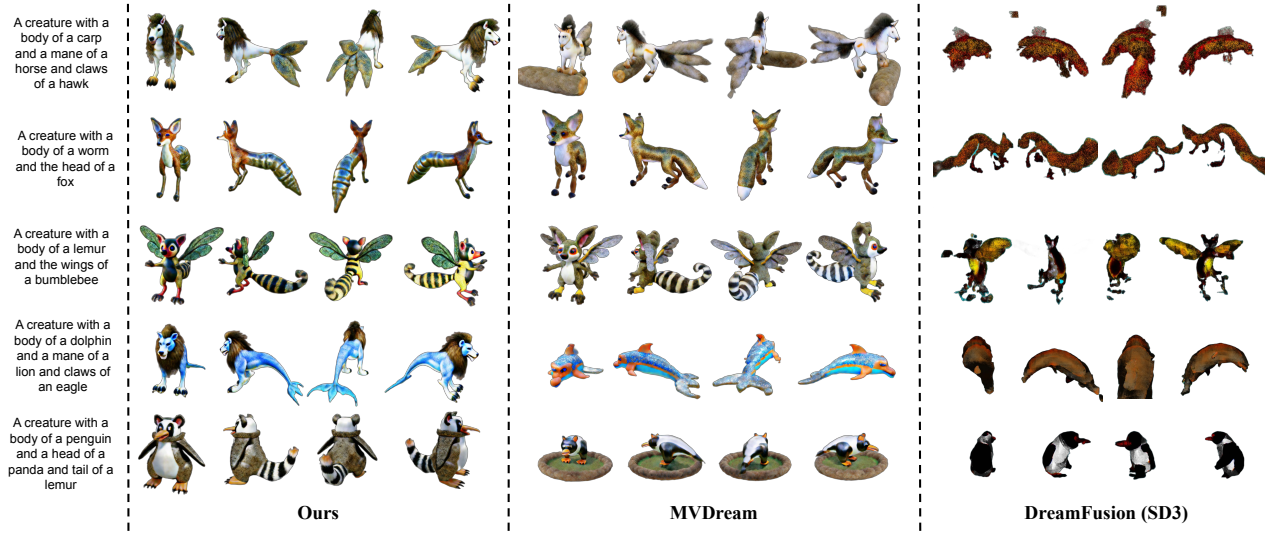

Figure 9. **Qualitative results of generated fantastic animals from different methods.** DreamBeastis capable of generating 3D assets with better part correspondence to part-specific prompts compared to MVDream or SD3. The fantastic animals created by MVDream and SD3 often either omit certain body parts or blend different animal elements globally, which is not the desired outcome.

results are shown in Table 2. We observe that DreamBeast consistently has higher similarity scores across CLIP types, indicating DreamBeast's better part-correspondence in generated 3D assets and the part-specific prompts.

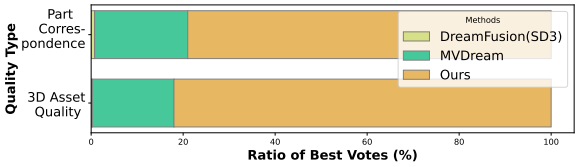

Figure 10. **User study results.** Participants were shown multi-view images generated from the same prompts and asked to select the best result. Our method receives significantly more votes for both part correspondence and overall 3D asset quality.

User Study Experiment. To assess the part correspondence quality and overall quality of the generated content, we asked users to compare our results with those from DreamFusion (SD3) and MVDream. We showed 24 users multi-view images generated from 20 random prompts and asked them to choose the best one. The results, shown in Figure 10, clearly demonstrate that DreamBeast outperforms existing methods. This suggests that our method effectively understands part compositions and generates high-quality 3D results.

Qualitative Results. As illustrated in Figure 9, DreamBeast demonstrates the ability to generate part-aware animals, with each relevant part closely adhering to its description. In contrast, MVDream produces assets with globally mixed animal features. Similarly, Dreamfusion (SD3) struggles to generate part-aware results, and both its 3D shape and texture quality are lacking, as discussed in Section 3.1.

Ablation Study. We examined the impact of the number of views on the extracted Part-Affinity maps, as shown in Table 3. Increasing the number of views enhances the context of part affinity, enabling the optimized Part-Affinity NeRF to better interpolate under unseen camera poses. This results in more accurate part-knowledge integration during the SDS process, which improves the CLIP similarity score. Moreover, we observed that performance gains diminish beyond 64 views, suggesting that our method can effectively capture complete Part-Affinity information in 3D with a relatively small number of views.

5. Conclusion

This work incorporates Part-Affinity knowledge to address the challenges associated with a limited part-level understanding of SDS-based 3D asset generation methods. Our proposed DreamBeast exhibits high precision in generating 3D assets with detailed part components, outperforming existing techniques in terms of both quality and efficiency. This contributes to the advancement of the field of creative and complex 3D content creation, paving the way for the development of more detailed and imaginative digital worlds.

Acknowledgments. The authors would like to thank Paul Engstler for his insightful feedback on the manuscript.

References

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

[2] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. Attribit: content creation with semantic attributes. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 193–202, 2013. 4

[3] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 100–116. Springer, 2019. 2

[4] Minghao Chen, Iro Laina, and Andrea Vedaldi. Training-free layout control with cross-attention guidance. *arXiv preprint arXiv:2304.03373*, 2023. 4

[5] Yongwei Chen, Tengfei Wang, Tong Wu, Xingang Pan, Kui Jia, and Ziwei Liu. Comboverse: Compositional 3d assets creation using spatially-aware diffusion guidance. *arXiv preprint arXiv:2403.12409*, 2024. 3, 4

[6] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2024. 3

[7] Xinhua Cheng, Tianyu Yang, Jianan Wang, Yu Li, Lei Zhang, Jian Zhang, and Li Yuan. Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts, 2023. 3

[8] Seokju Cho, Heeseong Shin, Sunghwan Hong, Anurag Arnab, Paul Hongsuck Seo, and Seungryong Kim. Cat-seg: Cost aggregation for open-vocabulary semantic segmentation, 2024. 6

[9] 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. *arXiv preprint arXiv:2212.08051*, 2022. 3

[10] Dave Epstein, Allan Jabri, Ben Poole, Alexei A. Efros, and Aleksander Holynski. Diffusion self-guidance for controllable image generation. 2023. 4

[11] Dave Epstein, Ben Poole, Ben Mildenhall, Alexei A. Efros, and Aleksander Holynski. Disentangled 3d scene generation with layout learning, 2024. 3, 4

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 2, 3, 7

[13] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM transactions on graphics (TOG)*, 23(3):652–663, 2004. 4

[14] Gege Gao, Weiyang Liu, Anpei Chen, Andreas Geiger, and Bernhard Schölkopf. Graphdreamer: Compositional 3d scene synthesis from scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21295–21304, 2024. 4

[15] Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. *European Conference on Computer Vision (ECCV)*, 2024. 2, 7

[16] Yutong He, Ruslan Salakhutdinov, J. Zico Kolter, and Mona Lisa. Localized text-to-image generation for free via cross attention control. *ArXiv*, abs/2306.14636, 2023. 4

[17] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 4

[18] 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*, 2023. 2, 7

[19] Yukun Huang, Jianan Wang, Ailing Zeng, He Cao, Xianbiao Qi, Yukai Shi, Zheng-Jun Zha, and Lei Zhang. Dreamwaltz: Make a scene with complex 3d animatable avatars. *arXiv:2305.12529*, 2023. 3

[20] Tomas Jakab, Ruining Li, Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Farm3D: Learning articulated 3d animals by distilling 2d diffusion. *arXiv preprint arXiv:2304.10535*, 2023. 3

[21] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *Acm Transactions on Graphics (TOG)*, 31(4):1–11, 2012. 4

[22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 5

[23] Yunji Kim, Jiyoung Lee, Jin-Hwa Kim, Jung-Woo Ha, and Jun-Yan Zhu. Dense text-to-image generation with attention modulation. In *ICCV*, 2023. 4

[24] Zizhang Li, Dor Litvak, Ruining Li, Yunzhi Zhang, Tomas Jakab, Christian Rupprecht, Shangzhe Wu, Andrea Vedaldi, and Jiajun Wu. Learning the 3d fauna of the web. *arXiv preprint arXiv:2401.02400*, 2024. 3

[25] 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, 2023. 3

[26] Baorui Ma, Haoge Deng, Junsheng Zhou, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Geodream: Disentangling 2d and geometric priors for high-fidelity and consistent 3d generation. 2023. 7

[27] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360 reconstruction of any object from a single image. In *CVPR*, 2023. 3

[28] 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. In *ECCV*, 2020. 5, 7

[29] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 4

[30] Quynh Phung, Songwei Ge, and Jia-Bin Huang. Grounded text-to-image synthesis with attention refocusing. *ArXiv*, abs/2306.05427, 2023. 4

[31] Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. In *2024 International Conference on 3D Vision (3DV)*, pages 651–663. IEEE, 2024. 3, 4

[32] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 2, 3, 5, 7

[33] Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, and Bernard Ghanem. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 3

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 7

[35] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 2, 7

[36] Adriana Schulz, Ariel Shamir, David IW Levin, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Design and fabrication by example. *ACM Transactions on Graphics (TOG)*, 33 (4):1–11, 2014. 4

[37] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation, 2024. 2, 3, 5, 7

[38] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2

[39] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. *arXiv preprint arXiv:2310.16818*, 2023. 3

[40] Shuyang Sun, Runjia Li, Philip Torr, Xiuye Gu, and Siyang Li. Clip as rnn: Segment countless visual concepts without training endeavor. In *CVPR*, 2024. 6

[41] 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. 3

[42] Konstantinos Tertikas, Despoina Paschalidou, Boxiao Pan, Jeong Joon Park, Mikaela Angelina Uy, Ioannis Emiris, Yannis Avrithis, and Leonidas Guibas. Generating part-aware editable 3d shapes without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4466–4478, 2023. 4

[43] Alexander Vilesov, Pradyumna Chari, and Achuta Kadambi. Cg3d: Compositional generation for text-to-3d via gaussian splatting. *arXiv preprint arXiv:2311.17907*, 2023. 4

[44] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A. Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation, 2022. 2, 3

[45] Zhaoning Wang, Ming Li, and Chen Chen. Luciddreaming: Controllable object-centric 3d generation. *arXiv preprint arXiv:2312.00588*, 2023. 4

[46] 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*, 2023. 3

[47] Meng Wei, Xiaoyu Yue, Wenwei Zhang, Shu Kong, Xihui Liu, and Jiangmiao Pang. Ov-parts: Towards open-vocabulary part segmentation. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 6

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

[49] Shangzhe Wu, Ruining Li, Tomas Jakab, Christian Rupprecht, and Andrea Vedaldi. MagicPony: Learning articulated 3d animals in the wild. 2023. 3

[50] Mengde Xu, Zheng Zhang, Fangyun Wei, Yutong Lin, Yue Cao, Han Hu, and Xiang Bai. A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model, 2022. 6

[51] Zizheng Yan, Jiapeng Zhou, Fanpeng Meng, Yushuang Wu, Lingteng Qiu, Zisheng Ye, Shuguang Cui, Guanying Chen, and Xiaoguang Han. Dreamdissector: Learning disentangled text-to-3d generation from 2d diffusion priors. *arXiv preprint arXiv:2407.16260*, 2024. 3

[52] Taoran Yi, Jiemin Fang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussian splatting with point cloud priors. *arxiv:2310.08529*, 2023. 3

[53] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5826–5835, 2021. 2