Zero-Shot Text-Guided Object Generation with Dream Fields

Ajay Jain[1,2*] Ben Mildenhall[2] Jonathan T. Barron[2] Pieter Abbeel[1] Ben Poole[2]

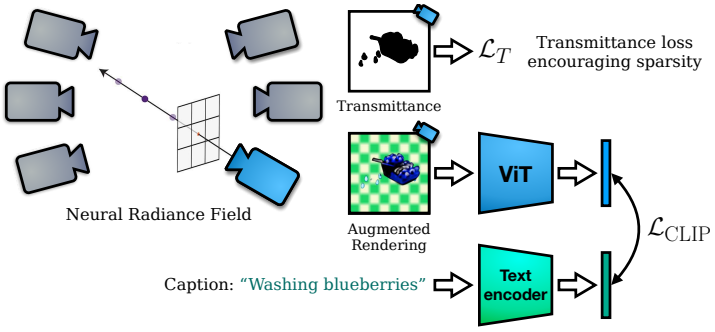

Figure 1. Given a caption, we learn a **Dream Field**, a continuous volumetric representation of an object's geometry and appearance learned with guidance from a pre-trained model. We optimize the Dream Field by rendering images of the object from random camera poses that are scored with *frozen* pre-trained **image** and **text** encoders trained on web images and alt-text. 2D views share the same underlying radiance field for consistent geometry.

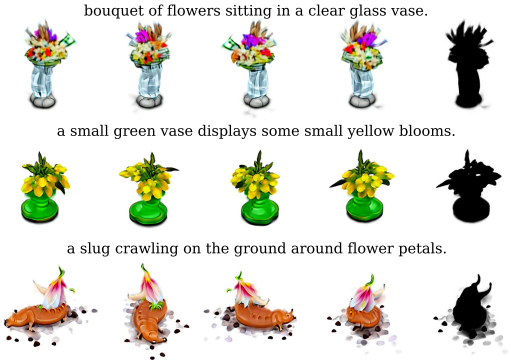

Figure 2. Example Dream Fields rendered from four perspectives. On the right, we show transmittance from the final perspective. We create diverse outputs using the compositionality of language; these captions from MSCOCO describe three flower arrangements with different properties like context and color.

Abstract

We combine neural rendering with multi-modal image and text representations to synthesize diverse 3D objects solely from natural language descriptions. Our method, Dream Fields, can generate the geometry and color of a wide range of objects without 3D supervision. Due to the scarcity of diverse, captioned 3D data, prior methods only generate objects from a handful of categories, such as ShapeNet. Instead, we guide generation with image-text models pre-trained on large datasets of captioned images from the web. Our method optimizes a Neural Radiance Field from many camera views so that rendered images score highly with a target caption according to a pre-trained CLIP model. To improve fidelity and visual quality, we introduce simple geometric priors, including sparsity-inducing transmittance regularization, scene bounds, and new MLP architectures. In experiments, Dream Fields produce realistic, multi-view consistent object geometry and color from a variety of natural language captions.

[1]UC Berkeley, [2]Google Research.
*Work done at Google.
Correspondence to ajayj@berkeley.edu.
Project website and code: https://ajayj.com/dreamfields

1. Introduction

Detailed 3D object models bring multimedia experiences to life. Games, virtual reality applications and films are each populated with thousands of object models, each designed and textured by hand with digital software. While expert artists can author high-fidelity assets, the process is painstakingly slow and expensive. Prior work leverages 3D datasets to synthesize shapes in the form of point clouds, voxel grids, triangle meshes, and implicit functions using generative models like GANs [4, 21, 57, 65]. These approaches only support a few object categories due to small labeled 3D shape datasets. But multimedia applications require a wide variety of content, and need both 3D geometry and texture.

In this work, we propose Dream Fields, a method to automatically generate open-set 3D models from natural language prompts. Unlike prior work, our method does not require any 3D training data, and uses natural language prompts that are easy to author with an expressive interface for specifying desired object properties. We demonstrate that the compositionality of language allows for flexible creative control over shapes, colors and styles.

A Dream Field is a Neural Radiance Field (NeRF) trained to maximize a deep perceptual metric with respect to both

the geometry and color of a scene. NeRF and other neural 3D representations have recently been successfully applied to novel view synthesis tasks where ground-truth RGB photos are available. NeRF is trained to reconstruct images from multiple viewpoints. As the learned radiance field is shared across viewpoints, NeRF can interpolate between viewpoints smoothly and consistently. Due to its neural representation, NeRF can be sampled at high spatial resolutions unlike voxel representations and point clouds, and are easy to optimize unlike explicit geometric representations like meshes as it is topology-free.

However, existing photographs are not available when creating novel objects from descriptions alone. Instead of learning to reconstruct known input photos, we learn a radiance field such that its renderings have high semantic similarity with a given text prompt. We extract these semantics with pre-trained neural image-text retrieval models like CLIP [46], learned from hundreds of millions of captioned images. As NeRF's volumetric rendering and CLIP's image-text representations are differentiable, we can optimize Dream Fields end-to-end for each prompt. Figure 1 illustrates our method.

In experiments, Dream Fields learn significant artifacts if we naively optimize the NeRF scene representation with textual supervision without adding additional geometric constraints (Figure 3). We propose general-purpose priors and demonstrate that they greatly improve the realism of results. Finally, we quantitatively evaluate open-set generation performance using a dataset of diverse object-centric prompts. Our contributions include:

- Using aligned image and text models to optimize NeRF without 3D shape or multi-view data,
- Dream Fields, a simple, constrained 3D representation with neural guidance that supports diverse 3D object generation from captions in zero-shot, and
- Simple geometric priors including transmittance regularization, scene bounds, and an MLP architecture that together improve fidelity.

2. Related Work

Our work is primarily inspired by DeepDream [34] and other methods for visualizing the preferred inputs and features of neural networks by optimizing in image space [36, 37, 39]. These methods enable the generation of interesting images from a pre-trained neural network without the additional training of a generative model. Closest to our work is [35], which studies differentiable image parameterizations in the context of style transfer. Our work replaces the style and content-based losses from that era with an image-text loss enabled by progress in contrastive representation learning on image-text datasets [12, 26, 46, 60]. The use of image-text models enables easy and flexible control over the style and content of generated imagery through textual prompt design. We optimize both geometry and color using

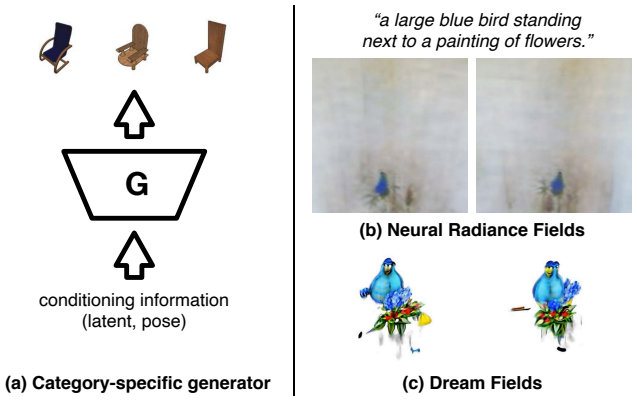

Figure 3. **Challenges of text-to-3D synthesis: (a) Poor generalization from limited 3D datasets:** Most 3D generative models are learned on datasets of specific object categories like ShapeNet [7], and won't generalize to novel concepts zero-shot. **(b) Neural Radiance Fields are too flexible without multi-view supervision:** NeRF learns to represent geometry and texture from scene-specific multi-view data, so it does not require a diverse dataset of objects. Yet, when only a source caption is available instead of multi-view images, NeRF produces significant artifacts (*e.g.*, near field occlusions). **(c) Dream Fields:** We introduce general geometric priors that retain much of NeRF's flexibility while improving realism.

the differentiable volumetric rendering and scene representation provided by NeRF, whereas [35] was restricted to fixed geometry and only optimized texture. Together these advances enable a fundamentally new capability: open-ended text-guided generation of object geometry and texture.

Concurrently to Dream Fields, a few early works have used CLIP [46] to synthesize or manipulate 3D object representations. CLIP-Forge [49] generates multiple object geometries from text prompts using a CLIP embedding-conditioned normalizing flow model and geometry-only decoder trained on ShapeNet categories. Still, CLIP-Forge generalizes poorly outside of ShapeNet categories and requires ground-truth multi-view images and voxel data. Text2Shape [9] learns a text-conditional Wasserstein GAN [1, 19] to synthesize novel voxelized objects, but only supports finite resolution generation of individual ShapeNet categories. In [10], object geometry is optimized evolutionarily for high CLIP score from a single view then manually colored. ClipMatrix [25] edits the vertices and textures of human SMPL models [31] to create stylized, deformable humanoid meshes. [44] creates an interactive interface to edit signed-distance fields in localized regions, though they do not optimize texture or synthesize new shapes. Text-based manipulation of existing objects is complementary to us.

For images, there has been an explosion of work that leverages CLIP to guide image generation. Digital artist Ryan Murdock (@advadnoun) used CLIP to guide learning of the weights of a SIREN network [52], similar to NeRF

but without volume rendering and focused on image generation. Katherine Crowson (@rivershavewings) combined CLIP with optimization of VQ-GAN codes [16] and used diffusion models as an image prior [14]. Recent work from Mario Klingemann (@quasimondo) and [43] have shown how CLIP can be used to guide GAN models like StyleGAN [27]. Some works have optimized parameters of vector graphics, suggesting CLIP guidance is highly general [17, 23, 50]. These methods highlighted the surprising capacity of what image-text models have learned and their utility for guiding 2D generative processes. Direct text to image synthesis with generative models has also improved tremendously in recent years [48, 61], but requires training large generative models on large-scale datasets, making such methods challenging to directly apply to text to 3D where no such datasets exist.

There is also growing progress on generative models with NeRF-based generators trained solely from 2D imagery. However, these models are category-specific and trained on large datasets of mostly forward-facing scenes [5, 20, 38, 51, 66], lacking the flexibility of open-set text-conditional models. Shape-agnostic priors have been used for 3D reconstruction [2, 58, 64].

3. Background

Our method combines Neural Radiance Fields (NeRF) [33] with an image-text loss from [46]. We begin by discussing these existing methods, and then detail our improved approach and methodology that enables high quality text to object generation.

3.1. Neural Radiance Fields

NeRF [33] parameterizes a scene's density and color using a multi-layer perceptron (MLP) with parameters θ trained with a photometric loss relying on multi-view photographs of a scene. In our simplified model, the NeRF network takes in a 3D position $\mathbf{x}$ and outputs parameters for an emission-absorption volume rendering model: density $\sigma_\theta(\mathbf{x})$ and color $\mathbf{c}_\theta(\mathbf{x})$. Images can be rendered from desired viewpoints by integrating color along an appropriate ray, $\mathbf{r}(t)$, for each pixel according to the volume rendering equation:

$$\mathbf{C}(r,\theta) = \int_{t_n}^{t_f} T(\mathbf{r},t)\sigma_\theta(\mathbf{r}(t))\mathbf{c}_\theta(\mathbf{r}(t))dt, \quad (1)$$

$$\text{where } T(\mathbf{r},\theta,t) = \exp\left(-\int_{t_n}^{t}\sigma_\theta(\mathbf{r}(s))ds\right). \quad (2)$$

The integral $T(\mathbf{r},\theta,t)$ is known as "transmittance" and describes the probability that light along the ray will not be absorbed when traveling from t_n (the near scene bound) to t. In practice [33], these two integrals are approximated by breaking up the ray into smaller segments $[t_{i-1}, t_i)$ within which σ and $\mathbf{c}$ are assumed to be roughly constant:

$$\mathbf{C}(\mathbf{r},\theta) \approx \sum_i T_i(1-\exp(-\sigma_\theta(\mathbf{r}(t_i))\delta_i))\mathbf{c}_\theta(\mathbf{r}(t_i)) \quad (3)$$

$$T_i = \exp\left(-\textstyle\sum_{j<i}\sigma_\theta(\mathbf{r}(t_j))\delta_j\right), \quad \delta_i = t_i - t_{i-1}. \quad (4)$$

For a given setting of MLP parameters θ and pose $\mathbf{p}$, we determine the appropriate ray for each pixel, compute rendered colors $C(\mathbf{r},\theta)$ and transmittances, and gather the results to form the rendered image, $I(\theta,\mathbf{p})$ and transmittance $T(\theta,\mathbf{p})$.

In order for the MLP to learn high frequency details more quickly [55], the input $\mathbf{x}$ is preprocessed by a sinusoidal positional encoding γ before being passed into the network:

$$\gamma(\mathbf{x}) = \left[\cos(2^l\mathbf{x}), \sin(2^l\mathbf{x})\right]_{l=0}^{L-1}, \quad (5)$$

where L is referred to as the number of "levels" of positional encoding. In our implementation, we specifically apply the integrated positional encoding (IPE) proposed in mip-NeRF to combat aliasing artifacts [3] combined with a random Fourier positional encoding basis [55] with frequency components sampled according to

$$\omega = 2^u\mathbf{d}, \quad \text{where } u \sim \mathcal{U}[0,L],\ \mathbf{d} \sim \mathcal{U}(\mathbb{S}^2). \quad (6)$$

3.2. Image-text models

Large-scale datasets of images paired with associated text have enabled training large-scale models that can accurately score whether an image and an associated caption are likely to correspond [12, 26, 46]. These models consist of an image encoder $\mathbf{g}$, and text encoder $\mathbf{h}$, that map images and text into a shared embedding space. Given a sentence y and an image $\mathbf{I}$, these image-text models produce a scalar score: $\mathbf{g}(\mathbf{I})^{\mathrm{T}}\mathbf{h}(\mathbf{y})$ that is high when the text is a good description of the image, and low when the the image and text are mismatched. Note that the embeddings $\mathbf{g}(\mathbf{I})$ and $\mathbf{h}(\mathbf{y})$ are often normalized, i.e. $\|\mathbf{g}(\mathbf{I})\| = \|\mathbf{h}(\mathbf{y})\| = 1$. Training is typically performed with a symmetric version of the InfoNCE loss [40, 45] that aims to maximize a variational lower bound on the mutual information between images and text. Prior work has shown that once trained, the image and text encoders are useful for a number of downstream tasks [46, 60]. In [48], the image and text encoders are used to score the correspondence of outputs of a generative image model to a target caption [48]. We build on this work by optimizing a volume to produce a high-scoring image, not just reranking.

4. Method

In this section, we develop Dream Fields: a zero-shot object synthesis method given only a natural language caption.

4.1. Object representation

Building on the NeRF scene representation (Section 3.1), a Dream Field optimizes an MLP with parameters θ that produces outputs $\sigma_\theta(\mathbf{x})$ and $\mathbf{c}_\theta(\mathbf{x})$ representing the differential

volume density and color of a scene at every 3D point $\mathbf{x}$. This field expresses object geometry via the density network. Our object representation is only dependent on 3D coordinates and not the camera's viewing direction, as we did not find it beneficial. Given a camera pose $\mathbf{p}$, we can render an image $\mathbf{I}(\theta, \mathbf{p})$ and compute the transmittance $T(\theta, \mathbf{p})$ using N segments via (4). Segments are spaced at roughly equal intervals with random jittering along the ray. The number of segments, N, determines the fidelity of the rendering. In practice, we fix it to 192 during optimization.

4.2. Objective

How can we train a Dream Field to represent a given caption? If we assume that an object can be described similarly when observed from any perspective, we can randomly sample poses and try to enforce that the rendered image matches the caption at all poses. We can implement this idea by using a CLIP network to measure the match between a caption and image given parameters θ and pose $\mathbf{p}$:

$$\mathcal{L}_{\text{CLIP}}(\theta, \text{pose } \mathbf{p}, \text{caption } y) = -\mathbf{g}(\mathbf{I}(\theta, \mathbf{p}))^{\text{T}}\mathbf{h}(\mathbf{y}) \quad (7)$$

where $\mathbf{g}(\cdot)$ and $\mathbf{h}(\cdot)$ are aligned representations of image and text semantics, and $\mathbf{I}(\theta, \mathbf{p})$ is a rendered image of the scene from camera pose $\mathbf{p}$. Each iteration of training, we sample a pose $\mathbf{p}$ from a prior distribution, render $\mathbf{I}$, and minimize $\mathcal{L}_{\text{CLIP}}$ with respect to the parameters of the Dream Field MLP, θ. Equation (7) measures the similarity of an image and the provided caption in feature space.

We primarily use image and text encoders from CLIP [46], which has a Vision Transformer image encoder $\mathbf{g}(\cdot)$ [15] and masked transformer text encoder $\mathbf{h}(\cdot)$ [56] trained contrastively on a large dataset of 400M captioned 224^2 images. We also use a baseline Locked Image-Text Tuning (LiT) ViT B/32 model from [60] trained via the same procedure as CLIP on a larger dataset of billions of higher-resolution (288^2) captioned images. The LiT training set was collected following a simplified version of the ALIGN web alt-text dataset collection process [26] and includes noisy captions.

Figure 1 shows a high-level overview of our method. DietNeRF [24] proposed a related semantic consistency regularizer for NeRF based on the idea that "a bulldozer is a bulldozer from any perspective". The method computed the similarity of a rendered and a real image. In contrast, (7) compares rendered images and a *caption*, allowing it to be used in zero-shot settings when there are no object photos.

4.3. Challenges with CLIP guidance

Due to their flexibility, Neural Radiance Fields are capable of high-fidelity novel view synthesis on a tremendous diversity of real-world scenes when supervised with multi-view consistent images. Their reconstruction loss will typically learn to remove artifacts like spurious density when sufficiently many input images are available. However, we find that the NeRF scene representation is too unconstrained when trained solely with $\mathcal{L}_{\text{CLIP}}$ (7) alone from a discrete set of viewpoints, resulting in severe artifacts that satisfy $\mathcal{L}_{\text{CLIP}}$ but are not visually compatible according to humans (see Figure 3b). NeRF learns high-frequency and near-field [62] artifacts like partially-transparent "floating" regions of density. It also fills the entire camera viewport rather than generating individual objects. Geometry is unrealistic, though textures reflect the caption, reminiscent of the artifacts in Deep Dream feature visualizations [34, 39].

Figure 4. To encourage coherent foreground objects, Dream Fields train with 3 types of background augmentations: blurred Gaussian noise, textures and checkerboards. At test time, we render with a white background. **Prompt:** *"A sculpture of a rooster."*

4.4. Pose sampling

Image data augmentations such as random crops are commonly used to improve and regularize image generation in DeepDream [34] and related work. Image augmentations can only use in-plane 2D transformations. Dream Fields support 3D data augmentations by sampling different camera pose extrinsics at each training iteration. We uniformly sample camera azimuth in 360° around the scene, so each training iteration sees a different orientation of the object. As the underlying scene representation is shared, this improves the realism of object geometry. For example, sampling azimuth in a narrow interval tended to create flat, billboard geometry.

The camera elevation, focal length and distance from the subject can also be augmented, but we did not find this necessary. Instead, we use a fixed camera focal length during optimization that is scaled by $m_{\text{focal}} = 1.2$ to enlarge the object 20%. Rendering cost is constant in the focal length.

4.5. Encouraging coherent objects through sparsity

To remove near-field artifacts and spurious density, we regularize the opacity of Dream Field renderings. Our best results maximize the average transmittance of rays passing through the volume up to a target constant. Transmittance is the probability that light along ray r is not absorbed by participating media when passing between point t along the ray and the near plane at t_n (2). We approximate the total transmittance along the ray as the joint probability of light passing through N discrete segments of the ray according to

Eq. (4). Then, we define the following transmittance loss:

$$\mathcal{L}_T = -\min(\tau, \text{mean}(T(\theta, \mathbf{p}))) \quad (8)$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CLIP}} + \lambda \mathcal{L}_T \quad (9)$$

This encourages a Dream Field to increase average transmittance up to a target transparency τ. We use $\tau = 88\%$ in experiments. τ is annealed in from $\tau = 40\%$ over 500 iterations to smoothly introduce transparency, which improves scene geometry and is essential to prevent completely transparent scenes. Scaling $1 - \tau \propto f^2/d^2$ preserves object cross sectional area for different focal and object distances.

When the rendering is alpha-composited with a simple white or black background during training, we find that the average transmittance approaches τ, but the scene is diffuse as the optimization populates the background. Augmenting the scene with random background images leads to coherent objects. Dream Fields use Gaussian noise, checkerboard patterns and the random Fourier textures from [35] as backgrounds. These are smoothed with a Gaussian blur with randomly sampled standard deviation. Background augmentations and a rendering during training are shown in Figure 4.

We qualitatively compare (9) to baseline sparsity regularizers in Figure 5. Our loss is inspired by the multiplicative opacity gating used by [35]. However, the gated loss has optimization challenges in practice due in part to its non-convexity. The simplified additive loss is more stable, and both are significantly sharper than prior approaches for sparsifying Neural Radiance Fields.

4.6. Localizing objects and bounding scene

When Neural Radiance Fields are trained to reconstruct images, scene contents will align with observations in a consistent fashion, such as the center of the scene in NeRF's Realistic Synthetic dataset [33]. Dream Fields can place density away from the center of the scene while still satisfying the CLIP loss as natural images in CLIP's training data will not always be centered. During training, we maintain an estimate of the 3D object's origin and shift rays accordingly. The origin is tracked via an exponential moving average of the center of mass of rendered density. To prevent objects from drifting too far, we bound the scene inside a cube by masking the density σ_θ.

4.7. Neural scene representation architecture

The NeRF network architecture proposed in [33] parameterizes scene density with a simple 8-layer MLP of constant width, and radiance with an additional two layers. We use a residual MLP architecture instead that introduces residual connections around every two dense layers. Within a residual block, we find it beneficial to introduce Layer Normalization at the beginning and increase the feature dimension in a bottleneck fashion. Layer Normalization improves optimization on challenging prompts. To mitigate vanishing gradient issues in highly transparent scenes, we replace ReLU activations with Swish [47] and rectify the predicted density σ_θ with a softplus function. Our MLP architecture uses 280K parameters per scene, while NeRF uses 494K parameters.

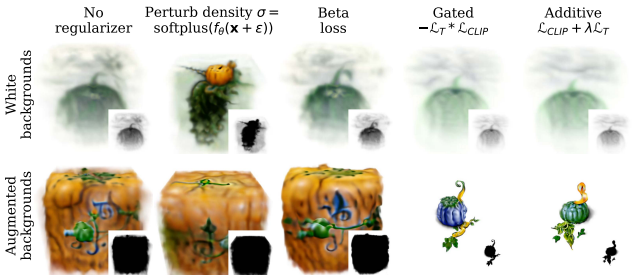

Figure 5. Our transmittance losses and background augmentations are complementary. **Top:** Without background augmentations, priors on transmittance (right three columns) do not remove low-density structures. NeRF's density perturbations improve coherence, but cloudy artifacts remain. **Bottom:** When the object is alpha composited with random backgrounds during training, CLIP fills the scene with opaque material to conceal the background. However, gated and our simplified additive transmittance regularizers both limit the opacity of the volume successfully and lead to a sharper object. Inset panels depict transmittance. **Prompt:** *"an illustration of a pumpkin on the vine."*

5. Evaluation

We evaluate the consistency of generated objects with their captions and the importance of scene representation, then show qualitative results and test whether Dream Fields can generalize compositionally. Ablations analyze regularizers, CLIP and camera poses. Finally, supplementary materials have further examples and videos.

5.1. Experimental setup

3D reconstruction methods are evaluated by comparing the learned geometry with a ground-truth reference model, *e.g.* with Chamfer Distance. Novel view synthesis techniques like LLFF [32] and NeRF do not have ground truth models, but compare renderings to pixel-aligned ground truth images from held-out poses with PSNR or LPIPS, a deep perceptual metric [63].

As we do not have access to diverse captioned 3D models or captioned multi-view data, Dream Fields are challenging to evaluate with geometric and image reference-based metrics. Instead, we use the CLIP R-Precision metric [41] from the text-to-image generation literature to measure how well rendered images align with the true caption. In the context of text-to-image synthesis, R-Precision measures the fraction of generated images that a retrieval model associates with the caption used to generate it. We use a different CLIP model for learning the Dream Field and computing the evaluation metric. As with NeRF evaluation, the image is rendered from

a held-out pose. Dream Fields are optimized with cameras at a 30° angle of elevation and evaluated at 45° elevation. For quantitative metrics, we render at resolution 168^2 during training as in [24]. For figures, we train with a 50% higher resolution of 252^2.

We collect an object-centric caption dataset with 153 captions as a subset of the Common Objects in Context (COCO) dataset [28] (see supplement for details). Object centric examples are those that have a single bounding box annotation and are filtered to exclude those captioned with certain phrases like "extreme close up". COCO includes 5 captions per image, but only one is used for generation. Hyperparameters were manually tuned for perceptual quality on a set of 20-74 distinct captions from the evaluation set, and are shared across all other scenes. Additional dataset details and hyperparameters are included in the supplement.

5.2. Analyzing retrieval metrics

In the absence of 3D training data, Dream Fields use geometric priors to constrain generation. To evaluate each proposed technique, we start from a simplified baseline Neural Radiance Field largely following [33] and introduce the priors one-by-one. We generate two objects per COCO caption using different seeds, for a total of 306 objects. Objects are synthesized with 10K iterations of CLIP ViT B/16 guided optimization of 168×168 rendered images, bilinearly upsampled to the contrastive model's input resolution for computational efficiency. R-Precision is computed with CLIP ViT B/32 [46] and LiT_{uu} B/32 [60] to measure the alignment of generations with the source caption.

Table 1 reports results. **The most significant improvements come from sparsity, scene bounds and architecture.** As an oracle, the ground truth images associated with object-centric COCO captions have high R-Precision. The NeRF representation converges poorly and introduces aliasing and banding artifacts, in part from its use of axis-aligned positional encodings.

We instead combine mip-NeRF's integrated positional encodings with random Fourier features, which improves qualitative results and removes a bias toward axis-aligned structures. However, the effect on precision is neutral or negative. The transmittance loss $\mathcal{L}_T$ in combination with background augmentations significantly improves retrieval precision $+18\%$ and $+15.6\%$, while the transmittance loss is not sufficient on its own. This is qualitatively shown in Figure 5. Our MLP architecture with residual connections, normalization, bottleneck-style feature dimensions and smooth nonlinearities further improves the R-Precision $+8\%$ and $+2\%$. Bounding the scene to a cube improves retrieval $+13\%$ and $+11\%$. The additional bounds explicitly mask density σ and concentrate samples along each ray.

We also scale up Dream Fields by optimizing with an image-text model trained on a larger captioned dataset of 3.6B images from [60]. We use a ViT B/32 model with image and text encoders trained from scratch. This corresponds to the uu configuration from [60], following the CLIP training procedure to learn both encoders contrastively. The LiT_{uu} ViT encoder used in our experiments takes higher resolution 288^2 images while CLIP is trained with 224^2 inputs. Still, LiT_{uu} B/32 is more compute-efficient than CLIP B/16 due to the larger patch size in the first layer.

	Method	R-Precision ↑ CLIP B/32	R-Precision ↑ LiT_{uu} B/32
Baseline	COCO GT images	77.1±3.4	75.2±3.5
	Simplified NeRF	31.4±2.7	10.8±1.8
Positional encoding	+ mip-NeRF IPE	29.7±2.6	12.4±1.9
	+ Higher freq. Fourier features	24.2±2.5	10.5±1.8
Sparsity, augment	+ random crops	25.8±2.5	10.5±1.8
	+ transmittance loss	23.7±2.4	7.6±1.5
	+ background aug.	44.1±2.8	26.1±2.5
Scene param.	+ MLP architecture	52.0±2.9	27.8±2.6
	+ scene bounds	65.4±2.7	**38.9±2.8**
	+ track origin	59.8±2.8	34.6±2.7
Scaling	+ LiT_{uu} ViT B/32	59.5±2.8	–
	+ 20K iterations, 252^2 renders	**68.3±2.7**	–

Table 1. **When used together, geometric priors improve caption retrieval precision.** We start with a simplified version of the NeRF scene representation and add in one prior at a time until all are used in conjunction. Captions are retrieved from rendered images of the generated objects at held-out camera poses using CLIP's ViT B/32. Objects are generated with $\mathcal{L}_{\text{CLIP}}$ guidance from the pre-trained CLIP ViT B/16 except in scaling experiments where we experiment with the higher-resolution LiT_{uu} B/32 model.

LiT_{uu} does not significantly help R-Precision when optimizing Dream Fields with low resolution renderings, perhaps because the CLIP B/32 model used for evaluation is trained on the same dataset as the CLIP B/16 model in earlier rows. Optimizing for longer with higher resolution 252^2 renderings closes the gap. LiT_{uu} improves visual quality and sharpness (Appendix A), suggesting that improvements in multimodal image-text models transfer to 3D generation.

5.3. Compositional generation

In Figure 6, we show non-cherrypicked generations that test the compositional generalization of Dream Fields to fine-grained variations in captions taken from the website of [48]. We independently vary the object generated and stylistic descriptors like shape and materials. DALL-E [48] also had a remarkable ability to combine concepts in prompts out of distribution, but was limited to 2D image synthesis.

Figure 6. Compositional object generation. Dream Fields allow users to express specific artistic styles via detailed captions. **Top two rows:** Similar to text-to-image experiments in [48], we generate objects with the caption *"armchair in the shape of an avocado. armchair imitating avocado."* **Bottom:** Generations vary the texture of a single snail. Captions follow the template *"a snail made of baguette. a snail with the texture of baguette"* Results are not cherry-picked.

Method	Loss or parameterization	R-Prec.
No regularizer	$\mathcal{L}_{\text{CLIP}}$ (7)	35.3
Perturb σ [33]	$\sigma = \text{softplus}(f_\theta(\mathbf{x}) + \epsilon)$	47.7
Beta prior [30]	(10)	50.3
Gated T [35]	$-\text{mean}(T(\theta, \mathbf{p})) \cdot \mathcal{L}_{\text{CLIP}}$	34.6
Clipped gated T	$-\mathcal{L}_T \cdot \mathcal{L}_{\text{CLIP}}$ (11)	**62.1**
Clipped additive T	$\mathcal{L}_{\text{CLIP}} + \lambda\mathcal{L}_T$ (9)	**62.1**

Table 2. Ablating sparsity regularizers. Optimization is done for 10K iterations at 168^2 resolution with LiT_{uu} ViT B/32 and background augmentation, and retrieval uses CLIP ViT B/32. For the purposes of ablation, we run one seed per caption (153 runs).

Dream Fields produces compositions of concepts in 3D, and supports fine-grained variations in prompts across several categories of objects. Some geometric details are not realistic, however. For example, generated snails have eye stalks attached to their shell rather than body, and the generated green vase is blurry.

5.4. Model ablations

Ablating sparsity regularizers While we regularize the mean transmittance, other sparsity losses are possible. We compare unregularized Dream Fields, perturbations to the density σ [33], regularization with a beta prior on transmittance [30], multiplicative gating versions of $\mathcal{L}_T$ and our additive $\mathcal{L}_T$ regularizer in Figure 5. On real-world scenes, NeRF added Gaussian noise to network predictions of the density prior to rectification as a regularizer. This can encourage sharper boundary definitions as small densities will often be zeroed by the perturbation. The beta prior from

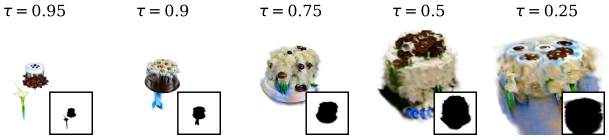

Figure 7. The target transmittance τ affects the size of generated objects. Inset panels depict transmittance. **Prompt from Object Centric COCO:** *"A cake toped* [sic] *with white frosting flowers with chocolate centers."*

Neural Volumes [30] encourages rays to either pass through the volume or be completely occluded:

$$\mathcal{L}^{\text{beta}}_{\text{total}} = \mathcal{L}_{\text{CLIP}} + \lambda \cdot \text{mean}(\log T(\theta, \mathbf{p}) + \log(1 - T(\theta, \mathbf{p}))) \quad (10)$$

The multiplicative loss is inspired by the opacity scaling of [35] for feature visualization. We scale the CLIP loss by a clipped mean transmittance:

$$\mathcal{L}_{\text{total}} = \min(\tau,\ \text{mean}(T(\theta, \mathbf{p}))) \cdot \mathcal{L}_{\text{CLIP}} \quad (11)$$

Table 2 compares the regularizers, showing that density perturbations and the beta prior improve R-Precision +12.4% and +15%, respectively. Scenes with clipped mean transmittance regularization best align with their captions, +26.8% over the baseline. The beta prior can fill scenes with opaque material even without background augmentations as it encourages both high and low transmittance. Multiplicative gating works well when clipped to a target and with background augmentations, but is also non-convex and sensitive to hyperparameters. Figure 7 shows the effect of varying the target transmittance τ with an additive loss.

Optimized model	Retrieval model R-Precision CLIP B/32	CLIP B/16	LiT_{uu} B/32
COCO GT	77.1±3.4	79.1±3.3	75.2±3.5
CLIP B/32 [46]	*(86.6±2.0)*	74.2±2.5	42.8±2.8
CLIP B/16 [46]	59.8±2.8	*(93.5±1.4)*	35.6±2.7
LiT_{uu} B/32	59.5±2.8	66.7±2.7	*(88.9±1.8)*

Table 3. The aligned image-text representation used to optimize Dream Fields influences their quantitative validation R-Precision according to a held-out retrieval model. All contrastive models produce high retrieval precision, though qualitatively CLIP B/32 produced overly smooth and simplified objects. We optimize for 10K iterations at 168^2 resolution. *(Italicized)* metrics use the optimized model at a held-out pose and indicate Dream Fields overfit.

Varying the image-text model We compare different image and text representations $h(\cdot), g(\cdot)$ used in $\mathcal{L}_{\text{CLIP}}$ (7) and for retrieval metrics. Table 3 shows the results. CLIP B/32, B/16 and LiT_{uu} B/32 all have high retrieval precision, indicating they can synthesize objects generally aligned with the provided captions. CLIP B/32 performs the best, outperforming the more compute intensive CLIP B/16 model. The architectures differ in the number of pixels encoded in each token supplied to the Transformer backbone, *i.e.* the ViT patch size. A larger patch size may be sufficient due to the low resolution of renders: 168^2 cropped to 154^2, then upsampled to CLIP's input size of 224^2. Qualitatively, training with LiT_{uu} B/32 produced the most detailed geometry and textures, suggesting that open-set evaluation is challenging.

Varying optimized camera poses Each training iteration, Dream Fields samples a camera pose $\mathbf{p}$ to render the scene. In experiments, we used a full 360° sampling range for the camera's azimuth, and fixed the elevation. Figure 8 shows multiple views of a bird when optimizing with smaller azimuth ranges. In the left-most column, a view from the central azimuth (frontal) is shown, and is realistic for all training configurations. Views from more extreme angles (right, left, rear view columns) have artifacts when the Dream Field is optimized with narrow azimuth ranges. Training with diverse cameras is important for viewpoint generalization.

6. Discussion and limitations

There are a number of limitations in Dream Fields. Generation requires iterative optimization, which can be expensive. 2K-20K iterations are sufficient for most objects, but more detail emerges when optimizing longer. Meta-learning [54] or amortization [42] could speed up synthesis.

We use the same prompt at all perspectives. This can lead to repeated patterns on multiple sides of an object. The target caption could be varied across different camera poses. Many of the prompts we tested involve multiple subjects,

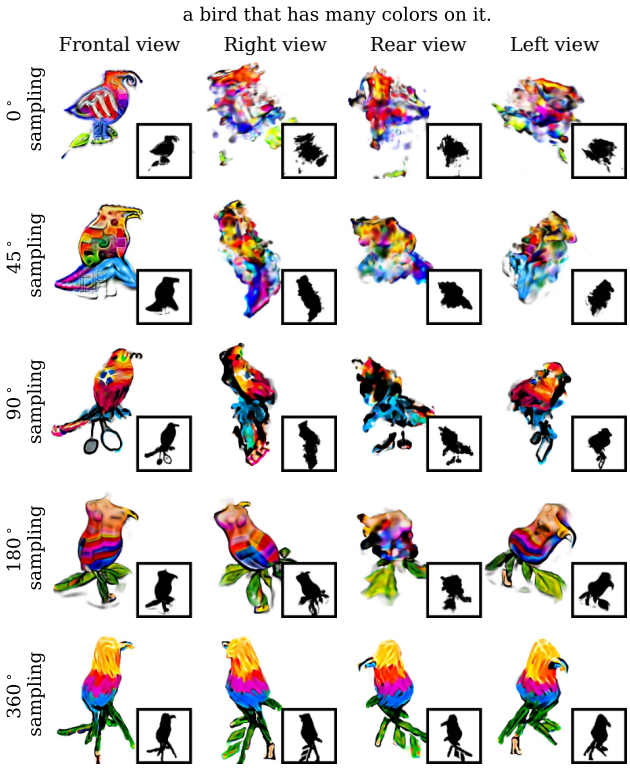

Figure 8. Training with diversely sampled camera poses improves generalization across views. In the top row, we sample camera azimuth from a single viewpoint. The rendered view from the same perspective (left column) is realistic, but the object structure is poor as seen from other angles. Qualitative results improve with larger sampling intervals, with the best results from 360° sampling.

but we do not target complex scene generation [6, 8, 11, 13] partly because CLIP poorly encodes spatial relations [29, 53]. Scene layout could be handled in a post-processing step.

The image-text models we use to score renderings are not perfect even on ground truth training images, so improvements in image-text models may transfer to 3D generation. Our reliance on pre-trained models inherits their harmful biases. Identifying methods that can detect and remove these biases is an important direction if these methods are to be useful for larger-scale asset generation.

7. Conclusion

Our work has begun to tackle the difficult problem of object generation from text. By combining scalable multimodal image-text models and multi-view consistent differentiable neural rendering with simple object priors, we are able to synthesize both geometry and color of 3D objects across a large variety of real-world text prompts. The language interface allows users to control the style and shape of the results, including materials and categories of objects, with easy-to-author prompts. We hope these methods will enable rapid asset creation for artists and multimedia applications.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. *ICML*, 2017. 2

[2] Jonathan T. Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *TPAMI*, 2015. 3

[3] 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. *ICCV*, 2021. 3

[4] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. *ECCV*, 2020. 1

[5] Eric Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. *CVPR*, 2021. 3

[6] Angel Chang, Will Monroe, Manolis Savva, Christopher Potts, and Christopher D. Manning. Text to 3d scene generation with rich lexical grounding. *ACL-IJCNLP*, 2015. 8

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

[8] Angel X Chang, Manolis Savva, and Christopher D Manning. Learning spatial knowledge for text to 3D scene generation. *EMNLP*, 2014. 8

[9] Kevin Chen, Christopher B. Choy, Manolis Savva, Angel X. Chang, Thomas A. Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *CoRR*, abs/1803.08495, 2018. 2

[10] Eric Chu. Evolving evocative 2d views of generated 3d objects. *NeurIPS Creativity and Design Workshop*, 2021. 2

[11] Bob Coyne and Richard Sproat. Wordseye: an automatic text-to-scene conversion system. *Computer graphics and interactive techniques*, 2001. 8

[12] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. *CVPR*, 2021. 2, 3

[13] Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W. Taylor, and Joshua M. Susskind. Unconstrained scene generation with locally conditioned radiance fields. *arXiv*, 2021. 8

[14] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv:2105.05233*, 2021. 3

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 4

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. *CVPR*, 2021. 3

[17] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, 2021. 3

[18] Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. https://distill.pub/2021/multimodal-neurons. 13

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

[20] Jiatao Gu, Lingjie Liu, Peng Wang, and Christian Theobalt. Stylenerf: A style-based 3d-aware generator for high-resolution image synthesis, 2021. 3

[21] Kunal Gupta and Manmohan Chandraker. Neural mesh flow: 3d manifold mesh generationvia diffeomorphic flows, 2020. 1

[22] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. 12

[23] Ajay Jain. VectorAscent: Generate vector graphics from a textual description, 2021. 3

[24] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. *ICCV*, 2021. 4, 6

[25] Nikolay Jetchev. Clipmatrix: Text-controlled creation of 3d textured meshes, 2021. 2

[26] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. *ICML*, 2021. 2, 3, 4

[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. *CVPR*, 2020. 3

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. *ECCV*, 2014. 6

[29] Nan Liu, Shuang Li, Yilun Du, Joshua B. Tenenbaum, and Antonio Torralba. Learning to compose visual relations. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 8

[30] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 2019. 7

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *SIGGRAPH Asia*, 2015. 2

[32] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 2019. 5

[33] 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. *ECCV*, 2020. 3, 5, 6, 7

[34] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015. 2, 4

[35] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 2018. https://distill.pub/2018/differentiable-parameterizations. 2, 5, 7

[36] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. *CVPR*, 2017. 2

[37] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *NeurIPS*, 2016. 2

[38] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. *CVPR*, 2021. 3

[39] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 2, 4

[40] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv:1807.03748*, 2018. 3

[41] Dong Huk Park, Samaneh Azadi, Xihui Liu, Trevor Darrell, and Anna Rohrbach. Benchmark for compositional text-to-image synthesis. *NeurIPS*, 2021. 5

[42] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *CVPR*, 2019. 8

[43] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *ICCV*, 2021. 3

[44] Victor Perez, Joel Simon, and Tal Shiri. Sculpting with words. 2021. 2

[45] Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. *ICML*, 2019. 3

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 2, 3, 4, 6, 8

[47] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. *CoRR*, abs/1710.05941, 2017. 5

[48] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 3, 6, 7

[49] Aditya Sanghi, Hang Chu, Joseph G. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation, 2021. 2

[50] Peter Schaldenbrand, Zhixuan Liu, and Jean Oh. Styleclipdraw: Coupling content and style in text-to-drawing synthesis, 2021. 3

[51] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *NeurIPS*, 2020. 3

[52] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 2

[53] Sanjay Subramanian, Will Merrill, Trevor Darrell, Matt Gardner, Sameer Singh, and Anna Rohrbach. Reclip: A strong zero-shot baseline for referring expression comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, Dublin, Ireland, May 2022. Association for Computational Linguistics. 8

[54] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. *CVPR*, 2021. 8

[55] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 3

[56] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, pages 5998–6008, 2017. 4

[57] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 1

[58] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning 3D Shape Priors for Shape Completion and Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[59] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *arXiv:2106.12052*, 2021. 13

[60] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning, 2021. 2, 3, 4, 6

[61] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-modal contrastive learning for text-to-image generation. *CVPR*, 2021. 3

[62] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 4

[63] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. *CVPR*, 2018. 5

[64] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to Reconstruct Shapes From Unseen Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[65] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. *ICCV*, 2021. 1

[66] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. CIPS-3D: A 3D-Aware Generator of GANs Based on Conditionally-Independent Pixel Synthesis. 2021. 3