Zero-Shot Text-Guided Object Generation with Dream Fields

Ajay Jain\textsuperscript{1,2}\textsuperscript{*} Ben Mildenhall\textsuperscript{2} Jonathan T. Barron\textsuperscript{2} Pieter Abbeel\textsuperscript{1} Ben Poole\textsuperscript{2}

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.

\textsuperscript{1}UC Berkeley, \textsuperscript{2}Google Research. 
\textsuperscript{*}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

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.
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 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 and trained on large datasets of mostly forward-facing 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 x and outputs parameters for an emission-absorption volume rendering model: density σθ(x) and color cθ(x). Images can be rendered from desired viewpoints by integrating color along an appropriate ray, r(t), for each pixel according to the volume rendering equation:

\[ C(r, \theta) \approx \sum_i T_i (1 - \exp(-\sigma(\theta)(r(t_i))\delta_i))c\theta(r(t_i)) \]  

(3)

\[ T_i = \exp\left(-\sum_{j<i} \sigma(\theta)(r(t_j))\delta_j\right) , \delta_i = t_i - t_{i-1} . \]  

(4)

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

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

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

(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

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

(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 g, and text encoder h, that map images and text into a shared embedding space. Given a sentence y and an image I, these image-text models produce a scalar score: g(I)h(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 g(I) and h(y) are often normalized, i.e. \|g(I)\| = \|h(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 σθ(x) and cθ(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 \( I(\theta, \mathbf{p}) \) 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 \( \theta \) and pose \( \mathbf{p} \):

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

where \( g(\cdot) \) and \( h(\cdot) \) are aligned representations of image and text semantics, and \( I(\theta, \mathbf{p}) \) 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 \( I \), and minimize \( \mathcal{L}_{\text{CLIP}} \) with respect to the parameters of the Dream Field MLP, \( \theta \). 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 \( g(\cdot) \) [15] and masked transformer text encoder \( h(\cdot) \) [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 \( \mathcal{L}_{\text{CLIP}} \) 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].

### 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 \( n_{\text{focal}} = 1.2 \) 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

![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.”](image-url)
Eq. (4). Then, we define the following transmittance loss:

\[
L_T = -\min(\tau, \text{mean}(T(\theta, p)))
\]

\[
L_{\text{total}} = L_{\text{CLIP}} + \lambda L_T
\]

This encourages a Dream Field to increase average transmittance up to a target transparency \(\tau\). We use \(\tau = 88\%\) in experiments. \(\tau\) 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 \(\tau\), 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 \(\sigma_\theta\).

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 \(\sigma_\theta\) with a softplus function. Our MLP architecture uses 280K parameters per scene, while NeRF uses 494K parameters.

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 168² during training as in [24]. For figures, we train with a 50% higher resolution of 252².

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 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 $L_T$ 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 $\sigma$ 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 288² images while CLIP is trained with 224² inputs. Still, LiT uu B/32 is more compute-efficient than CLIP B/16 due to the larger patch size in the first layer.

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 252² 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.

Table 2. Ablating sparsity regularizers. Optimization is done for 10K iterations at 168^2 resolution with LiT_uViT 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).

<table>
<thead>
<tr>
<th>Method</th>
<th>Loss or parameterization</th>
<th>R-Prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No regularizer</td>
<td>L_{CLIP} (7)</td>
<td>35.3</td>
</tr>
<tr>
<td>Perturb σ [33]</td>
<td>σ = softplus(fθ(x) + ε)</td>
<td>47.7</td>
</tr>
<tr>
<td>Beta prior [30]</td>
<td>(10)</td>
<td>50.3</td>
</tr>
<tr>
<td>Gated T [35]</td>
<td>− mean(T(θ, p)) · L_{CLIP}</td>
<td>34.6</td>
</tr>
<tr>
<td>Clipped gated T</td>
<td>− L_{T} · L_{CLIP} (11)</td>
<td>62.1</td>
</tr>
<tr>
<td>Clipped additive T</td>
<td>L_{CLIP} + λL_{T} (9)</td>
<td>62.1</td>
</tr>
</tbody>
</table>

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 \(\tau\) with an additive loss.

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 \(L_T\) and our additive \(L_T\) 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

\[
L_{\text{beta total}} = L_{\text{CLIP}} + \lambda \text{mean} \left( \log T(\theta, p) + \log(1 - T(\theta, p)) \right)
\]  

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

\[
L_{\text{total}} = \min(\tau, \text{mean}(T(\theta, p))) \cdot L_{\text{CLIP}}
\]

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

Figure 7. The target transmittance \(\tau\) affects the size of generated objects. Inset panels depict transmittance. Prompt from Object Centric COCO: “A cake topped [sic] with white frosting flowers with chocolate centers.”
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 168² 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 \( L_{\text{CLIP}} \) (7) and for retrieval metrics. Table 3 shows the results. CLIP B/32, B/16 and LiT\textsubscript{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: 168² cropped to 154², then upsampled to CLIP's input size of 224². Qualitatively, training with LiT\textsubscript{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, 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 multi-modal 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


[21] 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.


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


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


[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


[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


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

Zero-Shot Text-Guided Object Generation with Dream Fields
Supplementary Material

A. Qualitative results and ablations

An explanatory video with more qualitative results, code, an interactive Colab notebook, and object-centric prompts are available at https://ajayj.com/dreamfields. The video includes 360° renderings where the camera orbits the object, as well as associated depth maps.

Changing the image-text model Figure 9 qualitatively compares generations using guidance from three different contrastive image-text models. Dream Fields generated with LiTuu B/32 are generally sharper than those with CLIP models, but all three can produce objects reflecting some aspects of the prompts.

Diversity of synthesized objects In creative applications, users often want to select between multiple synthesized results. Dream Fields can synthesize multiple objects from the same prompt by changing the random seed before optimization. The seed changes the initialization of the NeRF weights, the camera pose sampled each iteration and the random background and crop augmentations. Figure 10 shows the effect of changing the seed for four object-centric COCO prompts. Changing the seed changes scene shape and layout. For example, the bus on the left is compressed into a cubical shape, while the bus on the right is elongated. Colors and textures are often similar across the seeds, though can also vary. Changing the seed is another dimension of control in addition to prompt engineering.

B. Object Centric COCO captions dataset

Our Object Centric COCO dataset includes 153 test set prompts and 74 development set prompts. Several additional prompts are used for qualitative results, and are included in the main paper alongside figures. Captions are included along with code on the project website.

C. Hyperparameters and training setup

Positional encoding Our Fourier feature positional encodings use $L = 8$ frequency levels, while novel view synthesis applications with image supervision commonly use $L = 10$ to fit high-frequency details in photographs. Low-frequency ablations in Table 1 use $L = 6$, which can improve convergence in the absence of our other geometric priors.

Rendering Scenes are bounded to a cube with side length 2. The camera is sampled at a fixed radius of 4 units from the center of the cubical scene bounds and an elevation of $30^\circ$ above the equator. Near and far planes are set at $4 \pm \sqrt{3}$ units from the camera based on the minimum and maximum possible distance to the corners of the cube. During training, we sample 192 points along each ray, spaced uniformly and jittered with uniform noise. Rendered $168^2$ views are cropped to $154^2$ and
upsampled to CLIP’s input resolution for scenes where we compute qualitative metrics or 252^2 views are cropped to 224^2 for certain higher-quality visualizations. Crop sizes are selected to cover about 80% of the image area. At test time, we sample 512 points along the rays and render at a higher resolution equal to CLIP’s input size of 224^2 or LiT’s input size of 288^2 for computing R-Precision and 400^2 for visualizations.

**Optimization** MLP parameters are initialized with the Flax [22] defaults: LeCun normal weights and zero bias for linear layers, and unit scaling and zero bias for layer normalization. The MLP is optimized with Adam with $\epsilon = 10^{-5}$. Learning rate warms up exponentially from $10^{-5}$ to $10^{-4}$ over 1500 iterations, then is held constant. The camera origin is separately tracked with an exponential moving average with decay rate 0.999 of the center of mass of rendered density. Figure 11 visualizes different forms of the transmittance loss.

**Hyperparameter selection** Hyperparameters are manually tuned for visual quality on a development set of 74 object-centric COCO captions distinct from the test set reported in the paper. Most tuning is done on a smaller subset of 20 of the 74 captions, and hyperparameters are shared across all scenes.

**Hardware** Optimization is done on 8 preemptible TPU cores, and 10K iterations takes approximately 1 hour 12 minutes. This means each Dream Field costs approximately $3-4 to generate on Google Cloud, which is economical for applications. Training is bottlenecked by MLP inference and backpropagation during volumetric rendering, not CLIP.

### D. Pixel and Voxel Baselines

We implemented 2D image optimization with a total variation loss and generative prior (CLIP Guided Diffusion), as well as a 3D voxel baselines to replace NeRF in Fig. 12. All results for this ablation optimize CLIP ViT B/16.

The 2D image is an RGB$\alpha$ pixel grid, composited with random backgrounds during optimization similar to Dream Fields. Optimizing a single 2D RGB$\alpha$ image does not produce a multi-view consistent 3D object, so other viewpoints cannot be rendered. Even with transmittance and TV regularization, the resulting image is noisy.

The voxel grid stores 128^3 RGB and alpha values, interpolated trilinearly at ray sample points and composited without a neural network using the PyTorch3D library. Despite the transmittance loss, data augmentations and scene bounds, the voxel grid also has significant low-density artifacts. The voxel baseline has CLIP B/32 R-Precision 37.0%±3.9, while NeRF has 59.8%±2.8 (Tables 1, 3) with 16× fewer parameters, showing that the neural representation improves consistency with the input caption in a generalizable way. Using a hybrid representation with an explicit voxel grid followed by a smaller MLP head might improve computational efficiency of Dream Fields without degrading quality.

**Pixel optimization** + transmittance loss, + TV loss

**128^3 RGB$\alpha$ voxels** 8M parameters + transmittance loss

**NeRF (ours)** 0.5M parameters + transmittance loss

Figure 12. Optimizing a Neural Radiance Field scene representation (bottom) leads to fewer artifacts than optimizing an explicit single-view 2D image (top) or 3D voxel grid (middle), even when the explicit representations are regularized. Our MLP has 16× fewer parameters than the voxel grid, which may contribute to smoother, less noisy objects. We use CLIP B/16 for this experiment.
Figure 13. Long-run training and validation curves averaged over 79 hand-written prompts. Transmittance remains close to the target $\tau$ throughout training. Dream Fields overfit to the image-text representations used for optimization, so in quantitative experiments, we stop training at 10K iterations. The standard error of the mean is shaded.

E. Signed distance field parameterization

In early experiments, we learned scene density $\sigma$ with the VolSDF parameterization [59] $\sigma(x) = \alpha \Phi_{\beta}(-d_{\Omega}(x))$ where $d_{\Omega}(x)$ is a signed distance function implicitly defining the object surface and $\Phi$ is the CDF of the Laplace distribution. This allows normal vector prediction with autodifferentiation and could improve the quality of the surface extracted from the radiance field. Dream Fields successfully train with this alternate parameterization and produce visually compelling objects. The SDF Eikonal loss introduces an additional loss weight hyperparameter, which benefits from some tuning. Alternate 3D representations are an interesting avenue for future work.

F. Impact of optimization time

Dream Fields can overfit to the aligned image-text representation used for optimization. Figure 13 shows the training losses, $\mathcal{L}_{\text{CLIP}}$ and mean transmittance $\text{mean}(T(\theta, p))$, as well as the validation R-Precision. Objects are generated with LiT ViT B/32 guidance, and R-Precision is computed with a different contrastive image-text model, CLIP ViT B/32. Validation renderings are also done at a held-out elevation angle. Training loss continues to improve over long optimization trajectories, up to 10× longer than reported in the main paper. However, validation retrieval accuracy declines after 5-10K iterations. The metrics are averaged over 79 different hand-written captions that test fine-grained variations in wording and prompt engineering.

Qualitatively, additional details and hyper-realistic effects are added over the course of long runs, shown in our supplementary video. Some details are not realistic, like floating text related to the typographic attacks identified in [18].

More augmentations may help further regularize the optimization. These include more aggressive 2D image augmentations such as smaller random crops, and more 3D data augmentations including varying focal length, varying distance from the subject and varying elevation. 3D data augmentations are supported by our approach.