Hierarchical Text-Conditional Image Generation with CLIP Latents

Aditya Ramesh[*]
OpenAI
aramesh@openai.com

Prafulla Dhariwal[*]
OpenAI
prafulla@openai.com

Alex Nichol[*]
OpenAI
alex@openai.com

Casey Chu[*]
OpenAI
casey@openai.com

Mark Chen
OpenAI
mark@openai.com

Abstract

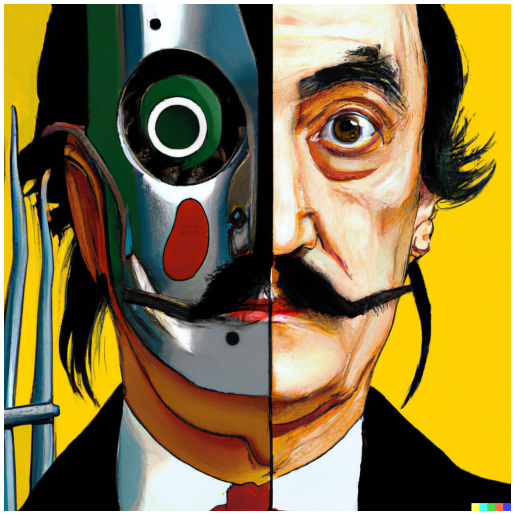
Contrastive models like CLIP have been shown to learn robust representations of images that capture both semantics and style. To leverage these representations for image generation, we propose a two-stage model: a prior that generates a CLIP image embedding given a text caption, and a decoder that generates an image conditioned on the image embedding. We show that explicitly generating image representations improves image diversity with minimal loss in photorealism and caption similarity. Our decoders conditioned on image representations can also produce variations of an image that preserve both its semantics and style, while varying the non-essential details absent from the image representation. Moreover, the joint embedding space of CLIP enables language-guided image manipulations in a zero-shot fashion. We use diffusion models for the decoder and experiment with both autoregressive and diffusion models for the prior, finding that the latter are computationally more efficient and produce higher-quality samples.

1 Introduction

Recent progress in computer vision has been driven by scaling models on large datasets of captioned images collected from the internet [10, 44, 60, 39, 31, 16]. Within this framework, CLIP [39] has emerged as a successful representation learner for images. CLIP embeddings have a number of desirable properties: they are robust to image distribution shift, have impressive zero-shot capabilities, and have been fine-tuned to achieve state-of-the-art results on a wide variety of vision and language tasks [45]. Concurrently, diffusion models [46, 48, 25] have emerged as a promising generative modeling framework, pushing the state-of-the-art on image and video generation tasks [11, 26, 24]. To achieve best results, diffusion models leverage a guidance technique [11, 24] which improves sample fidelity (for images, photorealism) at the cost of sample diversity.

In this work, we combine these two approaches for the problem of text-conditional image generation. We first train a diffusion *decoder* to invert the CLIP image *encoder*. Our inverter is non-deterministic, and can produce multiple images corresponding to a given image embedding. The presence of an encoder and its approximate inverse (the decoder) allows for capabilities beyond text-to-image translation. As in GAN inversion [62, 55], encoding and decoding an input image produces semantically similar output images (Figure 3). We can also interpolate between input images by inverting interpolations of their image embeddings (Figure 4). However, one notable advantage of using the CLIP latent space is the ability to semantically modify images by moving in the direction of any encoded text vector (Figure 5), whereas discovering these directions in GAN latent space involves

[*]Equal contribution



vibrant portrait painting of Salvador Dalí with a robotic half face

a shiba inu wearing a beret and black turtleneck

a close up of a handpalm with leaves growing from it

an espresso machine that makes coffee from human souls, artstation

panda mad scientist mixing sparkling chemicals, artstation

a corgi's head depicted as an explosion of a nebula

a dolphin in an astronaut suit on saturn, artstation

a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese

a teddy bear on a skateboard in times square

Figure 1: Selected 1024×1024 samples from a production version of our model.

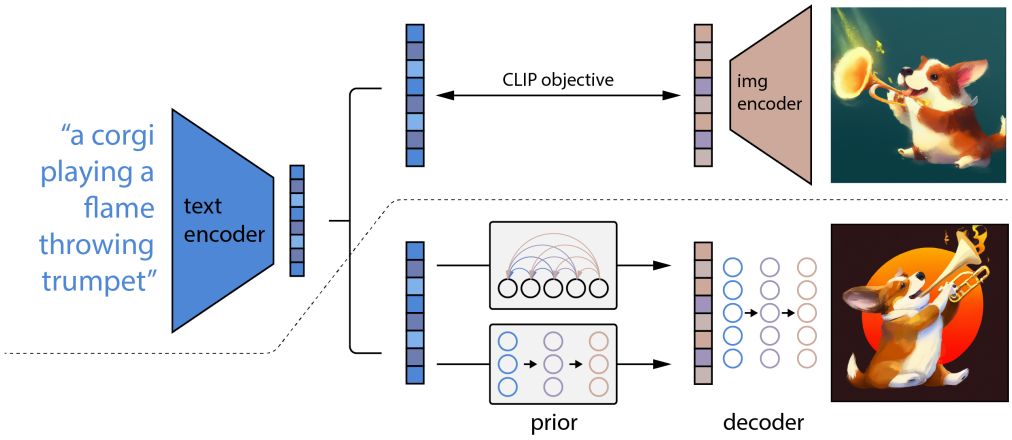


Figure 2: A high-level overview of unCLIP. Above the dotted line, we depict the CLIP training process, through which we learn a joint representation space for text and images. Below the dotted line, we depict our text-to-image generation process: a CLIP text embedding is first fed to an autoregressive or diffusion prior to produce an image embedding, and then this embedding is used to condition a diffusion decoder which produces a final image. Note that the CLIP model is frozen during training of the prior and decoder.

luck and diligent manual examination. Furthermore, encoding and decoding images also provides us with a tool for observing which features of the image are recognized or disregarded by CLIP.

To obtain a full generative model of images, we combine the CLIP image embedding *decoder* with a *prior* model, which generates possible CLIP image embeddings from a given text caption. We compare our text-to-image system with other systems such as DALL-E [40] and GLIDE [35], finding that our samples are comparable in quality to GLIDE, but with greater diversity in our generations. We also develop methods for training diffusion priors in latent space, and show that they achieve comparable performance to autoregressive priors, while being more compute-efficient. We refer to our full text-conditional image generation stack as *unCLIP*, since it generates images by inverting the CLIP image encoder.

2 Method

Our training dataset consists of pairs (x, y) of images x and their corresponding captions y. Given an image x, let z_i and z_t be its CLIP image and text embeddings, respectively. We design our generative stack to produce images from captions using two components:

- A *prior* $P(z_i|y)$ that produces CLIP image embeddings z_i conditioned on captions y.
- A *decoder* $P(x|z_i, y)$ that produces images x conditioned on CLIP image embeddings z_i (and optionally text captions y).

The decoder allows us to invert images given their CLIP image embeddings, while the prior allows us to learn a generative model of the image embeddings themselves. Stacking these two components yields a generative model $P(x|y)$ of images x given captions y:

$$P(x|y) = P(x, z_i|y) = P(x|z_i, y)P(z_i|y).$$

The first equality holds because z_i is a deterministic function of x. The second equality holds because of the chain rule. Thus, we can sample from the true conditional distribution $P(x|y)$ by first sampling z_i using the

prior, and then sampling x using the decoder. In the following sections, we describe our decoder and prior stacks. For training details and hyperparameters, refer to Appendix C.

2.1 Decoder

We use diffusion models [25, 48] to produce images conditioned on CLIP image embeddings (and optionally text captions). Specifically, we modify the architecture described in Nichol et al. (2021) by projecting and adding CLIP embeddings to the existing timestep embedding, and by projecting CLIP embeddings into four extra tokens of context that are concatenated to the sequence of outputs from the GLIDE text encoder. We retained the text conditioning pathway present in the original GLIDE model, hypothesizing that it could allow the diffusion model to learn aspects of natural language that CLIP fails to capture (e.g. variable binding), but find that it offers little help in this regard (Section 7).

While we can sample from the conditional distribution of the decoder directly, past work using diffusion models shows using guidance on the conditioning information [11, 24, 35] improves sample quality a lot. We enable classifier-free guidance [24] by randomly setting the CLIP embeddings to zero (or a learned embedding) 10% of the time, and randomly dropping the text caption 50% of the time during training.

To generate high resolution images, we train two diffusion upsampler models [34, 43]: one to upsample images from 64×64 to 256×256 resolution, and another to further upsample those to 1024×1024 resolution. To improve the robustness of our upsamplers, we slightly corrupt the conditioning images during training. For the first upsampling stage, we use gaussian blur [43], and for the second, we use a more diverse BSR degradation [42, 59]. To reduce training compute and improve numerical stability, we follow Rombach et al. [42] and train on random crops of images that are one-fourth the target size. We use only spatial convolutions in the model (i.e., no attention layers) and at inference time directly apply the model at the target resolution, observing that it readily generalizes to the higher resolution. We found no benefit from conditioning the upsamplers on the caption, and use unconditional ADMNets [11] with no guidance.

Blind image Super-Resolution

2.2 Prior

While a decoder can invert CLIP image embeddings z_i to produce images x, we need a prior model that produces z_i from captions y to enable image generations from text captions. We explore two different model classes for the prior model:

- *Autoregressive (AR)* prior: the CLIP image embedding z_i is converted into a sequence of discrete codes and predicted autoregressively conditioned on the caption y.
- *Diffusion* prior: The continuous vector z_i is directly modelled using a Gaussian diffusion model conditioned on the caption y.

In addition to the caption, we can condition the prior on the CLIP text embedding z_t since it is a deterministic function of the caption. To improve sample quality we also enable sampling using classifier-free guidance for both the AR and diffusion prior, by randomly dropping this text conditioning information 10% of the time during training.

To train and sample from the AR prior more efficiently, we first reduce the dimensionality of the CLIP image embeddings z_i by applying Principal Component Analysis (PCA) [37]. In particular, we find that the rank of the CLIP representation space is drastically reduced when training CLIP with SAM [15] while slightly improving evaluation metrics. We are able to preserve nearly all of the information[2] by retaining only 319 principal components out of the original 1,024. After applying PCA, we order the principal components by decreasing eigenvalue magnitude, quantize each of the 319 dimensions into 1,024 discrete buckets, and

特征值

[2]I.e., less than 1% average mean-squared error in reconstructing the image representations.

Figure 3: Variations of an input image by encoding with CLIP and then decoding with a diffusion model. The variations preserve both semantic information like presence of a clock in the painting and the overlapping strokes in the logo, as well as stylistic elements like the surrealism in the painting and the color gradients in the logo, while varying the non-essential details.

predict the resulting sequence using a Transformer [53] model with a causal attention mask. This results in a threefold reduction in the number of tokens predicted during inference, and improves training stability.

We condition the AR prior on the text caption and the CLIP text embedding by encoding them as a prefix to the sequence. Additionally, we prepend a token indicating the (quantized) dot product between the text embedding and image embedding, $z_i \cdot z_t$. This allows us to condition the model on a higher dot product, since higher text-image dot products correspond to captions which better describe the image. In practice, we find it beneficial to sample the dot product from the top half of the distribution.[3]

For the diffusion prior, we train a decoder-only Transformer with a causal attention mask on a sequence consisting of, in order: the encoded text, the CLIP text embedding, an embedding for the diffusion timestep, the noised CLIP image embedding, and a final embedding whose output from the Transformer is used to predict the unnoised CLIP image embedding. We choose not to condition the diffusion prior on $z_i \cdot z_t$ like in the AR prior; instead, we improve quality during sampling time by generating two samples of z_i and selecting the one with a higher dot product with z_t. Instead of using the ϵ-prediction formulation from Ho et al. [25], we find it better to train our model to predict the unnoised z_i directly, and use a mean-squared error loss on this prediction:

$$L_{\text{prior}} = \mathbb{E}_{t\sim[1,T],z_i^{(t)}\sim q_t}\left[\|f_\theta(z_i^{(t)}, t, y) - z_i\|^2\right]$$

[3] We swept over percentiles 50%, 70%, 85%, 95% and found 50% to be optimal in all experiments.

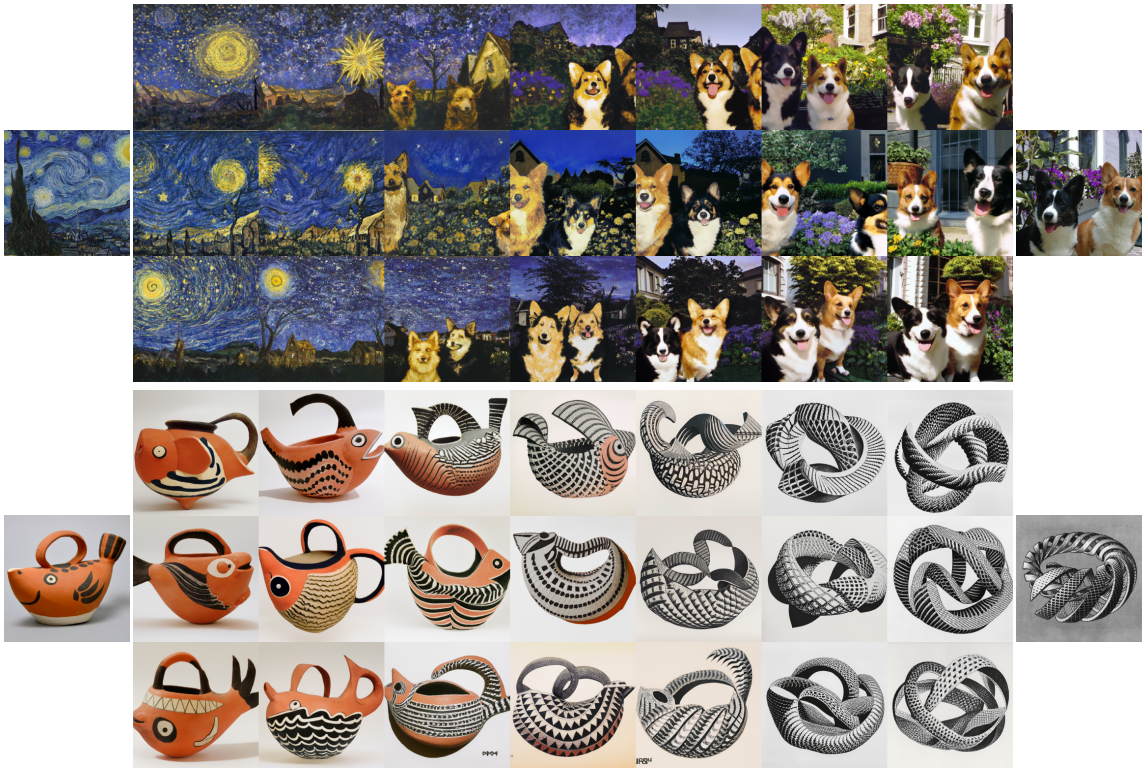


Figure 4: Variations between two images by interpolating their CLIP image embedding and then decoding with a diffusion model. We fix the decoder seed across each row. The intermediate variations naturally blend the content and style from both input images.

3 Image Manipulations

Our approach allows us to encode any given image x into a bipartite latent representation (z_i, x_T) that is sufficient for the decoder to produce an accurate reconstruction. The latent z_i describes the aspects of the image that are recognized by CLIP, while the latent x_T encodes all of the residual information necessary for the decoder to reconstruct x. The former is obtained by simply encoding the image with the CLIP image encoder. The latter is obtained by applying DDIM inversion (Appendix F in [11]) to x using the decoder, while conditioning on z_i. We describe three different kinds of manipulations that are enabled by this bipartite representation.

3.1 Variations

Given an image x, we can produce related images that share the same essential content but vary in other apects, such as shape and orientation (Figure 3). To do this, we apply the decoder to the bipartite representation (z_i, x_T) using DDIM with $\eta > 0$ for sampling. With $\eta = 0$, the decoder becomes deterministic and will reconstruct the given image x. Larger values of η introduce stochasticity into successive sampling steps, resulting in variations that are perceptually "centered" around the original image x. As η increases, these variations tell us what information was captured in the CLIP image embedding (and thus is preserved across samples), and what was lost (and thus changes across the samples).

a photo of a cat → an anime drawing of a super saiyan cat, artstation

a photo of a victorian house → a photo of a modern house

a photo of an adult lion → a photo of lion cub

a photo of a landscape in winter → a photo of a landscape in fall

Figure 5: Text diffs applied to images by interpolating between their CLIP image embeddings and a normalised difference of the CLIP text embeddings produced from the two descriptions. We also perform DDIM inversion to perfectly reconstruct the input image in the first column, and fix the decoder DDIM noise across each row.

3.2 Interpolations

It is also possible to blend two images x_1 and x_2 for variations (Figure 4), traversing all of the concepts in CLIP's embedding space that occur between them. To do this, we rotate between their CLIP embeddings z_{i_1} and z_{i_2} using spherical interpolation, yielding intermediate CLIP representations $z_{i_\theta} = \text{slerp}(z_{i_1}, z_{i_2}, \theta)$ as θ is varied from 0 to 1. There are two options for producing the intermediate DDIM latents along the trajectory. The first option involves interpolating between their DDIM inverted latents x_{T_1} and x_{T_2} (by setting $x_{T_\theta} = \text{slerp}(x_{T_1}, x_{T_2}, \theta)$), which yields a single trajectory whose endpoints reconstruct x_1 and x_2. The second option involves fixing the DDIM latent to a randomly-sampled value for all interpolates in the trajectory. This results in an infinite number of trajectories between x_1 and x_2, though the endpoints of these trajectories will generally no longer coincide with the original images. We use this approach in Figure 4.

3.3 Text Diffs

A key advantage of using CLIP compared to other models for image representations is that it embeds images and text to the same latent space, thus allowing us to apply language-guided image manipulations (i.e., text diffs), which we show in Figure 5. To modify the image to reflect a new text description y, we first obtain its CLIP text embedding z_t, as well as the CLIP text embedding z_{t_0} of a caption describing the current image[4]. We then compute a *text diff* vector $z_d = \text{norm}(z_t - z_{t_0})$ from these by taking their difference and

[4]Instead of a description of the current image, we also experimented with using a dummy caption like "a photo" for the baseline, or removing it altogether. These also worked well.

Granny Smith: 100%
iPod: 0%
Pizza: 0%

Granny Smith: 0.02%
iPod: 99.98%
Pizza: 0%

Granny Smith: 94.33%
iPod: 0%
Pizza: 5.66%

Figure 6: Variations of images featuring typographic attacks [20] paired with the CLIP model's predicted probabilities across three labels. Surprisingly, the decoder still recovers Granny Smith apples even when the predicted probability for this label is near 0%. We also find that our CLIP model is slightly less susceptible to the "pizza" attack than the models investigated in [20].

normalizing. Now, we can rotate between the image CLIP embedding z_i and the text diff vector z_d using spherical interpolation, yielding intermediate CLIP representations $z_\theta = \text{slerp}(z_i, z_d, \theta)$, where θ is increased linearly from 0 to a maximum value that is typically in $[0.25, 0.50]$. We produce the final outputs by decoding the interpolates z_θ, fixing the base DDIM noise to x_T throughout the entire trajectory.

4 Probing the CLIP Latent Space

Our decoder model provides a unique opportunity to explore CLIP latent space by allowing us to directly visualize what the CLIP image encoder is seeing. As an example use case, we can revisit cases where CLIP makes incorrect predictions, such as typographic attacks [20]. In these adversarial images, a piece of text is overlayed on top of an object, which causes CLIP to predict the object described by the text rather than the object depicted in the image. This piece of text essentially hides the original object in terms of output probabilities. In Figure 6, we show an example of this attack from [20], wherein an apple can be misclassified as an iPod. Surprisingly, we find that our decoder still generates pictures of apples with high probability even though the predicted probability of "Granny Smith" is near zero. Even more notable, the model never produces pictures of iPods, despite the very high relative predicted probability of this caption.

Figure 7: Visualization of reconstructions of CLIP latents from progressively more PCA dimensions (20, 30, 40, 80, 120, 160, 200, 320 dimensions), with the original source image on the far right. The lower dimensions preserve coarse-grained semantic information, whereas the higher dimensions encode finer-grained details about the exact form of the objects in the scene.

PCA reconstructions offer another tool for probing the structure of the CLIP latent space. In Figure 7, we take the CLIP image embeddings of a handful of source images and reconstruct them with progressively more PCA dimensions, and then visualize the reconstructed image embeddings using our decoder with DDIM on a fixed seed. This allows us to see what semantic information the different dimensions encode. We observe that the early PCA dimensions preserve coarse-grained semantic information such as what types of objects are in the scene, whereas the later PCA dimensions encode finer-grained detail such as the shapes and exact form of the objects. For example, in the first scene, the earlier dimensions seem to encode that there is food and perhaps a container present, whereas the later dimensions encode tomatoes and a bottle specifically. Figure 7 also serves as a visualization of what the AR prior is modeling, since the AR prior is trained to explicitly predict these principal components in this order.

5 Text-to-Image Generation

5.1 Importance of the Prior

Although we train a prior to generate CLIP image embeddings from captions, the prior is not strictly necessary for caption-to-image generation. For instance, our decoder can condition on both CLIP image embeddings and captions, but the CLIP image embedding is dropped 5% of the time during training in order to enable classifier-free guidance. Therefore, at sampling time, we can condition on only the caption, although this underperforms a model trained fully in this way (this model is GLIDE, and we do a thorough comparison with GLIDE in Sections 5.2 and 5.3). Another possibility is to feed the decoder the CLIP text embedding as if it were an image embedding, as previously observed [61, 54]. The first two rows of Figure 8 depicts samples obtained in these two ways; the third row depicts samples obtained with a prior. Conditioning the decoder on just the caption is clearly worst, but conditioning on text embeddings zero-shot does produce reasonable results. Building on this observation, another approach would be to train the decoder to condition on CLIP text embeddings [9] instead of CLIP image embeddings (although we would lose the capabilities mentioned in Section 4).

To quantify the effectiveness of these alternate approaches, we train two models: a small decoder conditioned on CLIP text embeddings, and a small unCLIP stack (diffusion prior and decoder). We then compare samples from the text-embedding decoder, samples from the unCLIP stack, and samples obtained from feeding text

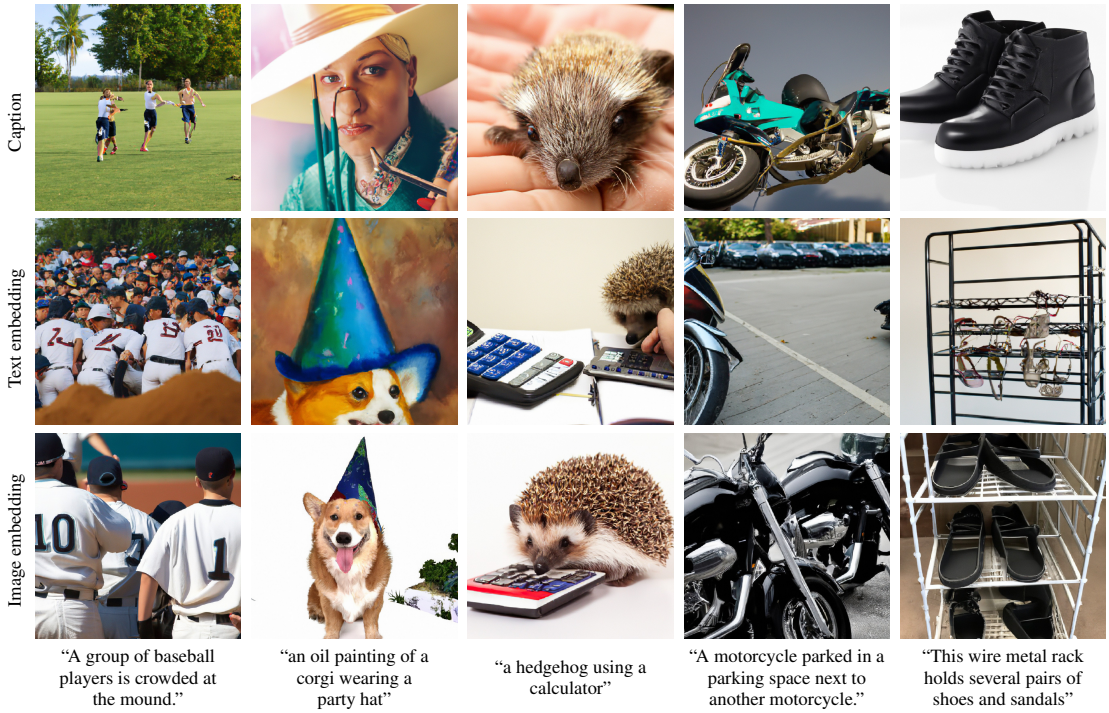


Figure 8: Samples using different conditioning signals for the *same* decoder. In the first row, we pass the text caption to the decoder, and pass a zero vector for the CLIP embedding. In the second row, we pass both the text caption and the CLIP text embedding of the caption. In the third row, we pass the text and a CLIP image embedding generated by an autoregressive prior for the given caption. Note that this decoder is only trained to do the text-to-image generation task (without the CLIP image representation) 5% of the time.

embeddings to the unCLIP decoder zero-shot, sweeping across guidance scales for all models. We find that these approaches respectively score FIDs of 9.16, 7.99, and 16.55 on a test set, suggesting the unCLIP approach is best. We also run human evaluations comparing the first two settings, sweeping over sampling hyperparameters for each using our human evaluation proxy model (Appendix A). We find that humans prefer the full unCLIP stack 57.0% $\pm$ 3.1% of the time for photorealism and 53.1% $\pm$ 3.1% of the time for caption similarity.

Given the importance of the prior, it is worth evaluating different approaches for training it. We compare both the AR and diffusion priors throughout our experiments. In all cases (Sections 5.2, 5.4, and 5.5), we find that the diffusion prior outperforms the AR prior for comparable model size and reduced training compute.

5.2 Human Evaluations

We observe in Figure 1 that unCLIP is capable of synthesizing complex, realistic images. While we can compare sample quality to past models using FID, it is not always aligned with human judgment. To better gauge the generation capabilities of our system, we conduct systematic human evaluations comparing unCLIP to GLIDE for photorealism, caption similarity, and sample diversity.

We follow the protocol of Ramesh et al., Nichol et al. [40, 35] for the first two evaluations: for photorealism, users are presented with pairs of images and must choose which looks more photorealistic; for caption

Figure 9: Samples when increasing guidance scale for both unCLIP and GLIDE, using the prompt, "A green vase filled with red roses sitting on top of table." For unCLIP, we fix the latent vectors sampled from the prior, and only vary the guidance scale of the decoder. For both models, we fix the diffusion noise seed for each column. Samples from unCLIP improve in quality (more realistic lighting and shadows) but do not change in content as we increase guidance scale, preserving semantic diversity even at high decoder guidance scales.

unCLIP Prior	Photorealism	Caption Similarity	Diversity
AR	$47.1\% \pm 3.1\%$	$41.1\% \pm 3.0\%$	$62.6\% \pm 3.0\%$
Diffusion	$48.9\% \pm 3.1\%$	$45.3\% \pm 3.0\%$	$70.5\% \pm 2.8\%$

Table 1: Human evaluations comparing unCLIP to GLIDE. We compare to both the AR and diffusion prior for unCLIP. Reported figures are 95% confidence intervals of the probability that the unCLIP model specified by the row beats GLIDE. Sampling hyperparameters for all models were swept to optimize an automated proxy for human photorealism evaluations.

similarity, users are additionally prompted with a caption, and must choose which image better matches the caption. In both evaluations, there is a third "Not sure" option. For diversity, we propose a new evaluation protocol in which humans are presented with two 4×4 grids of samples and must choose which is more diverse (with a third option, "Not sure"). For this evaluation, we produce sample grids using 1,000 captions from the MS-COCO validation set, and always compare sample grids for the same caption. Before running human comparisons, we swept over sampling hyperparameters for each model using a CLIP linear probe trained to be a proxy for human photorealism evaluations (Appendix A). These hyperparameters are fixed across all three types of evaluation.

We present our results in Table 1. In general, the diffusion prior performs better than the AR prior in pairwise comparisons against GLIDE. We find that humans still slightly prefer GLIDE to unCLIP in terms of photorealism, but the gap is very small. Even with similar photorealism, unCLIP is strongly preferred over GLIDE in terms of diversity, highlighting one of its benefits.

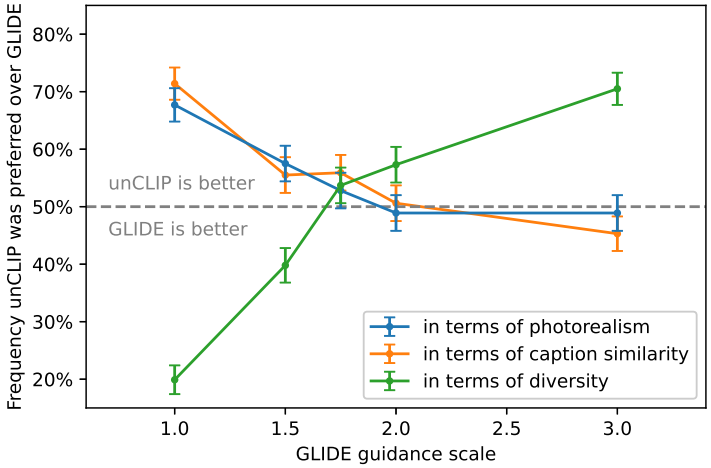


Figure 10: When comparing unCLIP (with our best sampling settings) to various settings of guidance scale for GLIDE, unCLIP was preferred by human evaluators on at least one axis among photorealism, caption similarity, and diversity for each comparison. At the higher guidance scales used to generate photorealistic images, unCLIP yields greater diversity for comparable photorealism and caption similarity.

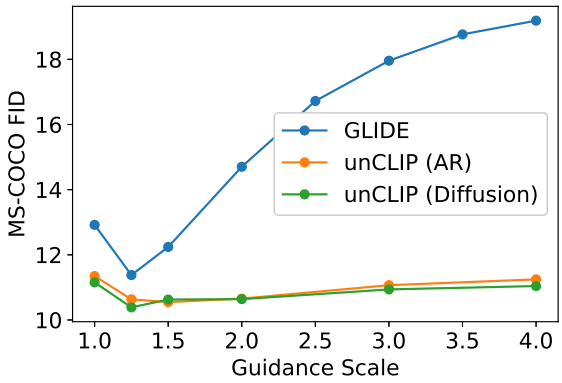


Figure 11: FID versus guidance scale for unCLIP and GLIDE. For the unCLIP priors, we swept over sampling hyperparameters and fixed to the settings with the best minimum FID.

5.3 Improved Diversity-Fidelity Trade-off with Guidance

Compared to GLIDE, we qualitatively observe that unCLIP is able to generate more diverse images while leveraging the guidance technique to improve sample quality. To understand why, consider Figure 9 where we increase guidance scale for both GLIDE and unCLIP. For GLIDE, the semantics (camera angle, color, size) converge as we increase guidance scale, whereas for unCLIP the semantic information of the scene is frozen in the CLIP image embedding and therefore does not collapse when guiding the decoder.

In Section 5.2, we observed that unCLIP achieves similar photorealism as GLIDE while maintaining more diversity, but that its caption matching capabilities were slightly worse. It is natural to ask whether GLIDE's guidance scale can be lowered to obtain the same diversity level as unCLIP while maintaining better caption

Model	FID	Zero-shot FID	Zero-shot FID (filt)
AttnGAN (Xu et al., 2017)	35.49		
DM-GAN (Zhu et al., 2019)	32.64		
DF-GAN (Tao et al., 2020)	21.42		
DM-GAN + CL (Ye et al., 2021)	20.79		
XMC-GAN (Zhang et al., 2021)	9.33		
LAFITE (Zhou et al., 2021)	8.12		
Make-A-Scene (Gafni et al., 2022)	**7.55**		
DALL-E (Ramesh et al., 2021)		~ 28	
LAFITE (Zhou et al., 2021)		26.94	
GLIDE (Nichol et al., 2021)		12.24	12.89
Make-A-Scene (Gafni et al., 2022)			11.84
unCLIP (AR prior)		10.63	11.08
unCLIP (Diffusion prior)		**10.39**	**10.87**

Table 2: Comparison of FID on MS-COCO 256×256. We use guidance scale 1.25 for the decoder for both the AR and diffusion prior, and achieve the best results using the diffusion prior.

matching. In Figure 10, we conduct a more careful study of this question by performing human evaluations across several GLIDE guidance scales. We find that GLIDE at guidance scale 2.0 is very close to the photorealism and caption similarity of unCLIP, while still producing less diverse samples.

Finally, in Figure 11 we compute MS-COCO zero-shot FID [23] while sweeping over guidance scale for both unCLIP and GLIDE, finding that guidance hurts the FID of unCLIP much less so than for GLIDE. In this evaluation, we fix the guidance scale of the unCLIP prior and only vary the guidance scale of the decoder. This is another indication that guidance hurts the diversity of GLIDE much more than unCLIP, since FID heavily penalizes non-diverse generations.

5.4 Comparison on MS-COCO

In the text-conditional image generation literature, it has become standard practice to evaluate FID on the MS-COCO [28] validation set. We present results on this benchmark in Table 2. Like GLIDE and DALL-E, unCLIP is not directly trained on the MS-COCO training set, but can still generalize to the validation set zero-shot. We find that, compared to these other zero-shot models, unCLIP achieves a new state-of-the-art FID of 10.39 when sampling with the diffusion prior. In Figure 12, we visually compare unCLIP to various recent text-conditional image generation models on several captions from MS-COCO. We find that, like the other methods, unCLIP produces realistic scenes that capture the text prompts.

5.5 Aesthetic Quality Comparison

We additionally perform automated aesthetic quality evaluations comparing unCLIP to GLIDE. Our goal with this evaluation is to assess how well each model produces artistic illustrations and photographs. To this end, we generated 512 "artistic" captions using GPT-3 [4] by prompting it with captions for existing artwork (both real and AI generated). Next, we trained a CLIP linear probe to predict human aesthetic judgments using the AVA dataset [33] (Appendix A). For each model and set of sampling hyperparameters, we produce four images for each prompt, and report the mean predicted aesthetic judgment over the full batch of 2048 images.

In Figure 13, we present results on our aesthetic quality evaluation. We find that guidance improves aesthetic quality for both GLIDE and unCLIP. For unCLIP, we only guide the decoder (we found that guiding the prior hurt results). We also plot the aesthetic quality against Recall[5], since guidance typically induces a trade-off

[5]Recall is computed with respect to the training dataset.

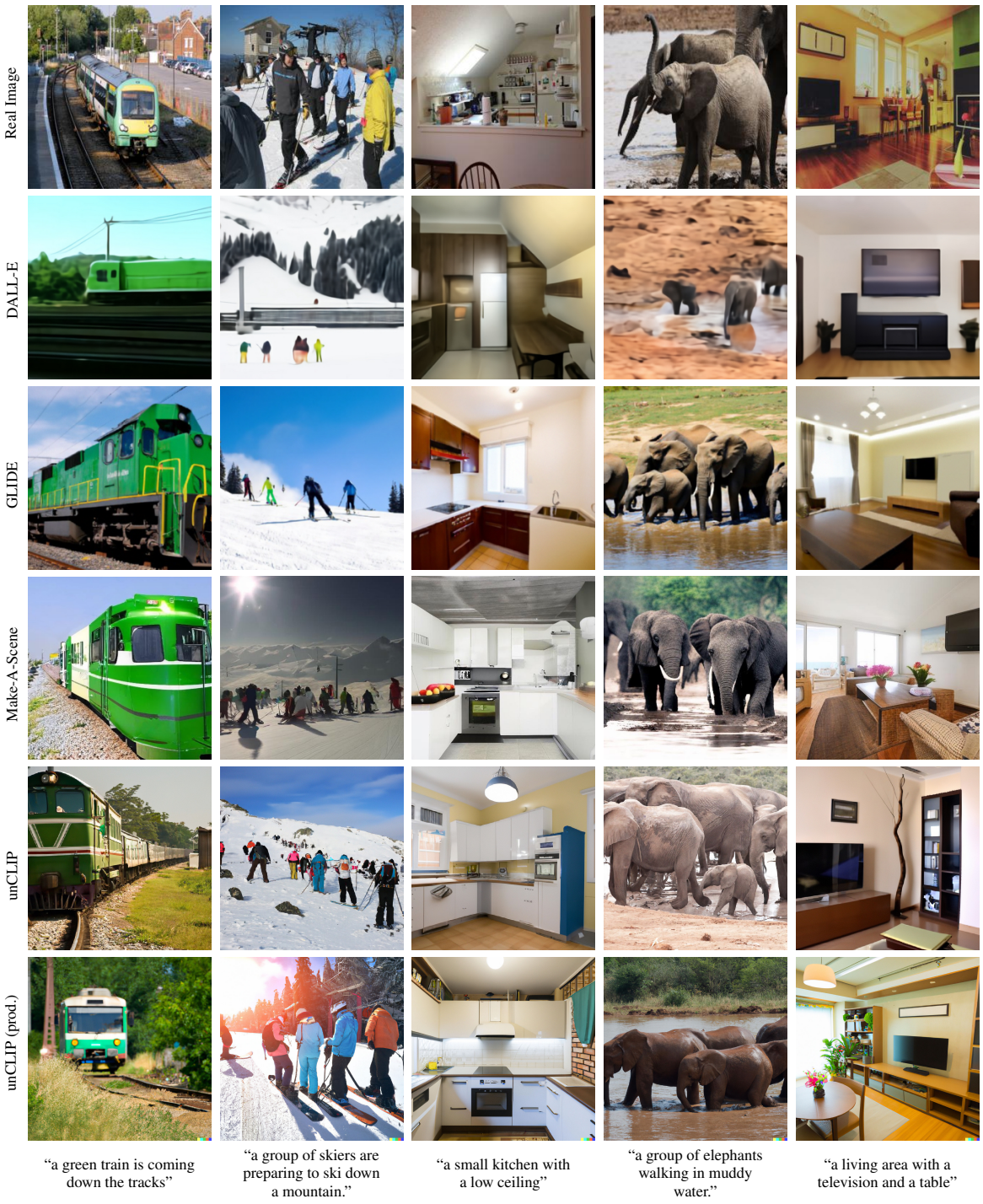


Figure 12: Random image samples on MS-COCO prompts.

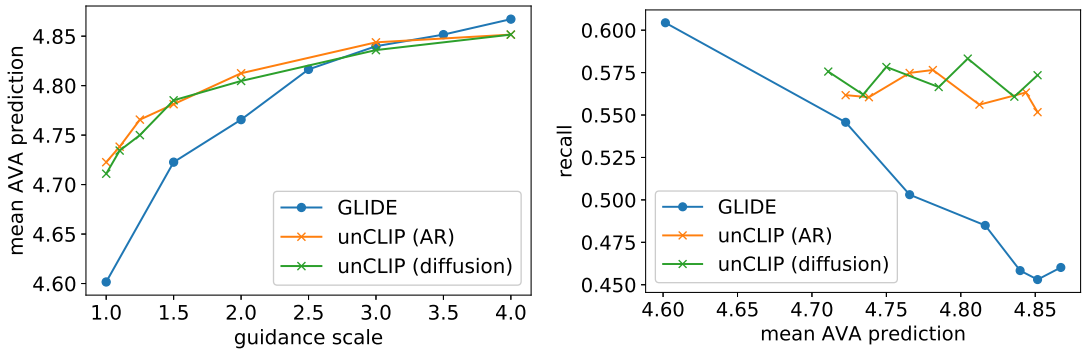


Figure 13: Aesthetic quality evaluations comparing GLIDE and unCLIP using 512 auto-generated artistic prompts. We find that both models benefit from guidance, but unCLIP does not sacrifice recall for aesthetic quality.

between fidelity and diversity. Interestingly, we find that guiding unCLIP does not decrease Recall while still improving aesthetic quality according to this metric.

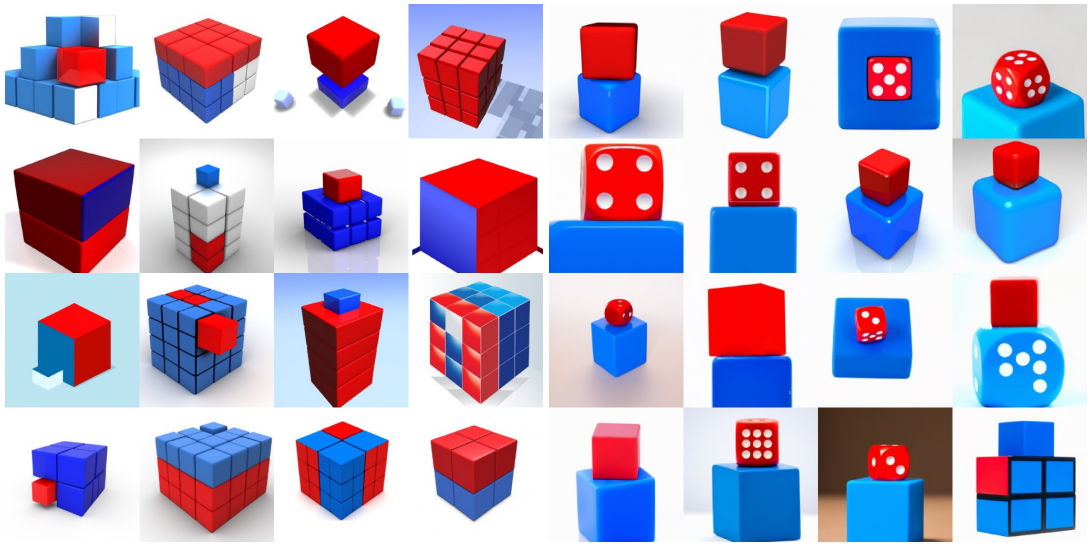
6 Related Work

Synthetic image generation is a well studied problem, and most popular techniques for unconditional image generation have also been applied to the text-conditional setting. Many previous works have trained GANs [21] on publicly available image captioning datasets to produce text-conditional image samples [56, 63, 49, 58, 57]. Other works have adapted the VQ-VAE approach [52] to text-conditional image generation by training autoregressive transformers on sequences of text tokens followed by image tokens [40, 12, 1]. Finally, some works have applied diffusion models to the problem, training either continuous [35] or discrete [22] diffusion models with auxiliary text encoders to handle textual input.

Previous works have leveraged hierarchical generative processes to create high-quality synthetic images. Razavi et al. [41] trains a multi-layer discrete autoencoder, allowing them to first sample coarse-grained latent codes and then use this as conditioning information when sampling higher-resolution latent codes. Child, Vahdat and Kautz [5, 50] generate images using VAEs with a hierarchy of latent codes that increase progressively with resolution. Concurrently with our work, Gafni et al. [17] conditions a generative image model on segmentation masks, allowing for a generative process that first samples a semantic map of an image and then conditions the generated image on this information.

The computational benefits of using diffusion to model a latent space has been noted by previous works. Preechakul et al. [38] propose an autoencoder framework where diffusion models are used to render latent variables as images, and a second diffusion model is used to generate these latents (similar to our diffusion prior). Vahdat et al. [51] use a score-based model for the latent space of a VAE, while Rombach et al. [42] use diffusion models on the latents obtained from a VQGAN [14] like autoencoder.

Since its release, CLIP [39] has been used extensively to steer generative image models towards text prompts. Galatolo et al., Patashnik et al., Murdock, Gal et al. [19, 36, 32, 18] guide GANs using gradients from a CLIP model. For diffusion models, Dhariwal and Nichol [11] introduced classifier guidance as a way to use gradients from a classifier trained on noised images to steer the model towards higher quality generations. Nichol et al. [35] train a CLIP model on noised images and guide a text-conditional diffusion model, while Crowson, Crowson [7, 8] use an unnoised CLIP model to guide unconditional or class-conditional diffusion models. Ho and Salimans [24] introduced classifier-free guidance and showed that one can perform guidance



(a) unCLIP (b) GLIDE

Figure 14: Samples from unCLIP and GLIDE for the prompt “a red cube on top of a blue cube”.

implictly from the predictions of the model with and without the conditioning information, thus removing the need for a classifier. Nichol et al. [35] showed classifier-free guidance works more favorably than CLIP guidance for text conditional image generation.

Several previous works have trained generative image models that are directly conditioned on CLIP embeddings. Zhou et al. [61] condition GAN models on randomly perturbed CLIP image embeddings, finding that these models can generalize to CLIP text embeddings to produce text-conditional images. Crowson [9] trained diffusion models conditioned on CLIP text embeddings, allowing for direct text-conditional image generation. Wang et al. [54] train an autoregressive generative model conditioned on CLIP image embeddings, finding that it generalizes to CLIP text embeddings well enough to allow for text-conditional image synthesis.

Bordes et al. [3] train diffusion models conditioned on image representations from contrastive models. While the diffusion models themselves cannot generate images unconditionally, the authors experimented with a simple approach for two-stage image generation by employing Kernel Density Estimation to sample image representations. By feeding these generated representations to the diffusion model, they can generate images end-to-end in a way similar to our proposed technique. However, our work differs from this in two ways: first, we use multimodal contrastive representations rather than image-only representations; second, we employ much more powerful generative models for the first stage of the generation hierarchy, and these generative models are conditioned on text.

7 Limitations and Risks

Although conditioning image generation on CLIP embeddings improves diversity, this choice does come with certain limitations. In particular, unCLIP is worse at binding attributes to objects than a corresponding GLIDE model. In Figure 14, we find that unCLIP struggles more than GLIDE with a prompt where it must bind two separate objects (cubes) to two separate attributes (colors). We hypothesize that this occurs because the CLIP embedding itself does not explicitly bind attributes to objects, and find that reconstructions from the decoder often mix up attributes and objects, as shown in Figure 15. A similar and likely related issue is that unCLIP

Figure 15: Reconstructions from the decoder for difficult binding problems. We find that the reconstructions mix up objects and attributes. In the first two examples, the model mixes up the color of two objects. In the rightmost example, the model does not reliably reconstruct the relative size of two objects.

Figure 16: Samples from unCLIP for the prompt, "A sign that says deep learning."

struggles at producing coherent text, as illustrated in Figure 16; it is possible that the CLIP embedding does not precisely encode spelling information of rendered text. This issue is likely made worse because the BPE encoding we use obscures the spelling of the words in a caption from the model, so the model needs to have independently seen each token written out in the training images in order to learn to render it.

We also note that our stack still has a hard time producing details in complex scenes (Figure 17). We hypothesize that this is a limitation of our decoder hierarchy producing an image at a base resolution of 64×64 and then upsampling it. Training our unCLIP decoder at a higher base resolution should be able to alleviate this, at the cost of additional training and inference compute.

As discussed in the GLIDE paper, image generation models carry risks related to deceptive and otherwise harmful content. unCLIP's performance improvements also raise the risk profile over GLIDE. As the technology matures, it leaves fewer traces and indicators that outputs are AI-generated, making it easier to mistake generated images for authentic ones and vice versa. More research is also needed on how the change in architecture changes how the model learns biases in training data.

(a) A high quality photo of a dog playing in a green field next to a lake.

(b) A high quality photo of Times Square.

Figure 17: unCLIP samples show low levels of detail for some complex scenes.

The risks of these models should be assessed in relation to the particular deployment context, which includes training data, guardrails in place, the deployment space, and who will have access. A preliminary analysis of these issues in the context of the DALL·E 2 Preview platform (the first deployment of an unCLIP model), can be found in Mishkin et al. [30].

8 Acknowledgements

We'd like to thank Jong Wook Kim, Hyeonwoo Noh, Alec Radford, Pranav Shyam, and Ilya Sutskever for helpful discussions and contributions to our work. We'd also like to thank Yunxin Jiao for creating several figures used in the paper. We are grateful to the Acceleration and Supercomputing teams at OpenAI for their work on software and hardware infrastructure this project used.

References

[1] Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. CM3: A Causal Masked Multimodal Model of the Internet. *arXiv:2201.07520*, 2022.

[2] Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models. *CoRR*, abs/2201.06503, 2022. URL `https://arxiv.org/abs/2201.06503`.

[3] Florian Bordes, Randall Balestriero, and Pascal Vincent. High Fidelity Visualization of What Your Self-Supervised Representation Knows About. *arXiv:2112.09164*, 2021.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165*, 2020.

[5] Rewon Child. Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images. *arXiv:2011.10650*, 2021.

[6] Katherine Crowson. AVA Linear Probe. `https://twitter.com/RiversHaveWings/status/1472346186728173568?s=20&t=T-HRr3Gw5HRGjQaMDtRe3A`, 2021.

[7] Katherine Crowson. CLIP guided diffusion HQ 256x256. `https://colab.research.google.com/drive/12a_Wrfi2_gwwAuN3VvMTwVMz9TfqctNj`, 2021.

[8] Katherine Crowson. CLIP Guided Diffusion 512x512, Secondary Model Method. `https://twitter.com/RiversHaveWings/status/1462859669454536711`, 2021.

[9] Katherine Crowson. v-diffusion. `https://github.com/crowsonkb/v-diffusion-pytorch`, 2021.

[10] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. *arXiv:2006.06666*, 2020.

[11] Prafulla Dhariwal and Alex Nichol. Diffusion Models Beat GANs on Image Synthesis. *arXiv:2105.05233*, 2021.

[12] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. CogView: Mastering Text-to-Image Generation via Transformers. *arXiv:2105.13290*, 2021.

[13] 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. *arXiv:2010.11929*, 2020.

[14] Patrick Esser, Robin Rombach, and Björn Ommer. Taming Transformers for High-Resolution Image Synthesis. *arXiv:2012.09841*, 2020.

[15] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-Aware Minimization for Efficiently Improving Generalization. *arXiv:2010.01412*, 2020.

[16] Andreas Fürst, Elisabeth Rumetshofer, Viet Thuong Tran, Hubert Ramsauer, Fei Tang, Johannes Lehner, D P Kreil, Michael K Kopp, Günter Klambauer, Angela Bitto-Nemling, and Sepp Hochreiter. CLOOB: Modern Hopfield Networks with InfoLOOB Outperform CLIP, 2022. URL https://openreview.net/forum?id=qw674L9PfQE.

[17] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-A-Scene: Scene-Based Text-to-Image Generation with Human Priors. *arXiv:2203.13131*, 2022.

[18] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *arXiv:2108.00946*, 2021.

[19] Federico A. Galatolo, Mario G. C. A. Cimino, and Gigliola Vaglini. Generating images from caption and vice versa via CLIP-Guided Generative Latent Space Search. *arXiv:2102.01645*, 2021.

[20] 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. doi: 10.23915/distill.00030. https://distill.pub/2021/multimodal-neurons.

[21] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661*, 2014.

[22] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector Quantized Diffusion Model for Text-to-Image Synthesis. *arXiv:2111.14822*, 2021.

[23] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[24] Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *arXiv:2006.11239*, 2020.

[26] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded Diffusion Models for High Fidelity Image Generation. *arXiv:2106.15282*, 2021.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv:1405.0312*, 2014.

[29] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv:1711.05101*, 2017.

[30] Pamela Mishkin, Lama Ahmad, Miles Brundage, Gretchen Krueger, and Girish Sastry. DALL·E 2 Preview - Risks and Limitations. 2022. URL https://github.com/openai/dalle-2-preview/blob/main/system-card.md.

[31] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. SLIP: Self-supervision meets Language-Image Pre-training. *arXiv:2112.12750*, 2021.

[32] Ryan Murdock. The Big Sleep. https://twitter.com/advadnoun/status/1351038053033406468, 2021.

[33] Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2415, 2012. doi: 10.1109/CVPR.2012.6247954.

[34] Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. *arXiv:2102.09672*, 2021.

[35] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models. *arXiv:2112.10741*, 2021.

[36] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery. *arXiv:2103.17249*, 2021.

[37] Karl Pearson. LIII. On lines and planes of closest fit to systems of points in space, November 1901. URL `https://doi.org/10.1080/14786440109462720`.

[38] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. *arXiv:2111.15640*, 2021.

[39] 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. *arXiv:2103.00020*, 2021.

[40] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. *arXiv:2102.12092*, 2021.

[41] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating Diverse High-Fidelity Images with VQ-VAE-2. *arXiv:1906.00446*, 2019.

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv:2112.10752*, 2021.

[43] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image Super-Resolution via Iterative Refinement. *arXiv:arXiv:2104.07636*, 2021.

[44] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning Visual Representations with Caption Annotations. *arXiv:2008.01392*, 2020.

[45] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How Much Can CLIP Benefit Vision-and-Language Tasks? *arXiv:2107.06383*, 2021.

[46] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. *arXiv:1503.03585*, 2015.

[47] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. *arXiv:2010.02502*, 2020.

[48] Yang Song and Stefano Ermon. Improved Techniques for Training Score-Based Generative Models. *arXiv:2006.09011*, 2020.

[49] Ming Tao, Hao Tang, Songsong Wu, Nicu Sebe, Xiao-Yuan Jing, Fei Wu, and Bingkun Bao. DF-GAN: Deep Fusion Generative Adversarial Networks for Text-to-Image Synthesis. *arXiv:2008.05865*, 2020.

[50] Arash Vahdat and Jan Kautz. NVAE: A Deep Hierarchical Variational Autoencoder. *arXiv:2007.03898*, 2020.

[51] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based Generative Modeling in Latent Space. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[52] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural Discrete Representation Learning. *arXiv:1711.00937*, 2017.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv:1706.03762*, 2017.

[54] Zihao Wang, Wei Liu, Qian He, Xinglong Wu, and Zili Yi. CLIP-GEN: Language-Free Training of a Text-to-Image Generator with CLIP. *arXiv:2203.00386*, 2022.

[55] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. GAN Inversion: A Survey. *arXiv:2101.05278*, 2021.

[56] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. *arXiv:1711.10485*, 2017.

[57] Hui Ye, Xiulong Yang, Martin Takac, Rajshekhar Sunderraman, and Shihao Ji. Improving Text-to-Image Synthesis Using Contrastive Learning. *arXiv:2107.02423*, 2021.

[58] Han Zhang, Jing Yu Koh, Jason Baldridge, Honglak Lee, and Yinfei Yang. Cross-Modal Contrastive Learning for Text-to-Image Generation. *arXiv:2101.04702*, 2021.

[59] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a Practical Degradation Model for Deep Blind Image Super-Resolution. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. doi: 10.1109/iccv48922.2021.00475. URL `http://dx.doi.org/10.1109/ICCV48922.2021.00475`.

[60] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D. Manning, and Curtis P. Langlotz. Contrastive Learning of Medical Visual Representations from Paired Images and Text. *arXiv:2010.00747*, 2020.

[61] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. LAFITE: Towards Language-Free Training for Text-to-Image Generation. *arXiv:2111.13792*, 2021.

[62] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative Visual Manipulation on the Natural Image Manifold. *arXiv:1609.03552*, 2016.

[63] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis. *arXiv:1904.01310*, 2019.

A Linear Probes for Evaluations

For our evaluations, we leverage two new linear probes on top of a CLIP ViT-L/14 [13] model. To automate aesthetic quality evaluations, we follow the procedure used by Crowson [6], training a linear regression model on images and mean ratings from the AVA dataset [33]. To reduce the cost of hyperparameter sweeps before conducting human evaluations, we train a logistic regression model to predict win probabilities between pairs of images. To train this model, we used 15,000 pairwise image comparisons gathered from all of our previous human evaluations. For each comparison i, we computed CLIP image embeddings x_i and y_i for the two images in the pair. We then trained a linear model $f(x)$ such that $1/(1 + \exp(f(x_i) - f(y_i)))$ approximates the probability that a human prefers the image for y_i. This can be reduced to a logistic regression problem with inputs equal to $y_i - x_i$.

B Error Bars for Human Evaluation

When computing error bars for human evaluations, we use the normal approximation interval with $p = 0.95$. We expect the normal approximation to be accurate for such a large sample size of $n = 1000$.

C Training Details

The unCLIP models used for the experiments in this paper were trained with the hyperparameters described below, unless otherwise noted. We additionally trained a production version of unCLIP using similarly sized models but with modified architectures and trained for longer; we include changes to accommodate product and safety requirements (e.g. inpainting, preventing unwanted memorization), and train on a larger dataset that is filtered for aesthetic quality and safety. We report model and training hyperparameters for the paper models in Table 3. All models were trained using Adam [27] with corrected weight decay [29] and momentum $\beta_1 = 0.9$.

Our CLIP model uses a ViT-H/16 [13] image encoder that consumes 256×256 resolution images, and has width 1280 with 32 Transformer [53] blocks. The text encoder also follows the architecture described in Radford et al. [39]: it is a Transformer [53] with a causal attention mask, with width 1024 and 24 Transformer blocks. Both models are trained with learning rate 3×10^{-4} and SAM [15] with $\rho = 0.1$, where the perturbations are applied independently by the replicas, each of which uses batch size 64. The remaining hyperparameters are the same as those reported in Radford et al. [39].

When training the encoder, we sample from the CLIP [39] and DALL-E [40] datasets (approximately 650M images in total) with equal probability. When training the decoder, upsamplers, and prior, we use only the DALL-E dataset [40] (approximately 250M images). Incorporating the noisier CLIP dataset while training the generative stack negatively impacted sample quality in our initial evaluations.

Our decoder architecture is the 3.5 billion parameter GLIDE model, with the same architecture and diffusion hyperparameters as in Nichol et al. [35]. We train with learned sigma and sample with 250 strided sampling steps as in Nichol and Dhariwal [34].

We use the ADMNet architecture [11] for the upsamplers. In the first upsampling stage, we use a cosine noising schedule, 320 channels and a depth of 3 resblocks per resolution inside the ADMNet. We also apply gaussian blur (kernel size 3, sigma 0.6) as described in Saharia et al. [43]. In the second upsampling stage, we use a linear noising schedule, 192 channels, a depth of 2 resblocks per resolution, and train with the BSR degradation from Rombach et al. [42]. Neither upsampler uses attention. To reduce inference time, we use DDIM [47] and manually tune the number of steps, with 27 steps for 256×256 model, and 15 steps for the 1024×1024 model.

For the AR prior, we use a Transformer text encoder with width 2048 and 24 blocks and a decoder with a causal attention mask, width 1664, and 24 blocks. For the diffusion prior, we use a Transformer with width 2048 and 24 blocks, and sample with Analytic DPM [2] with 64 strided sampling steps. To reuse hyperparameters tuned for diffusion noise schedules on images from Dhariwal and Nichol [11], we scale the CLIP embedding inputs by 17.2 to match the empirical variance of RGB pixel values of ImageNet images scaled to $[-1, 1]$.

	AR prior	Diffusion prior	64	$64 \rightarrow 256$	$256 \rightarrow 1024$
Diffusion steps	-	1000	1000	1000	1000
Noise schedule	-	cosine	cosine	cosine	linear
Sampling steps	-	64	250	27	15
Sampling variance method	-	analytic [2]	learned [34]	DDIM [47]	DDIM [47]
Crop fraction	-	-	-	0.25	0.25
Model size	1B	1B	3.5B	700M	300M
Channels	-	-	512	320	192
Depth	-	-	3	3	2
Channels multiple	-	-	1,2,3,4	1,2,3,4	1,1,2,2,4,4
Heads channels	-	-	64	-	-
Attention resolution	-	-	32,16,8	-	-
Text encoder context	256	256	256	-	-
Text encoder width	2048	2048	2048	-	-
Text encoder depth	24	24	24	-	-
Text encoder heads	32	32	32	-	-
Latent decoder context	384	-	-	-	-
Latent decoder width	1664	-	-	-	-
Latent decoder depth	24	-	-	-	-
Latent decoder heads	26	-	-	-	-
Dropout	-	-	0.1	0.1	-
Weight decay	4.0e-2	6.0e-2	-	-	-
Batch size	4096	4096	2048	1024	512
Iterations	1M	600K	800K	1M	1M
Learning rate	1.6e-4	1.1e-4	1.2e-4	1.2e-4	1.0e-4
Adam β_2	0.91	0.96	0.999	0.999	0.999
Adam ϵ	1.0e-10	1.0e-6	1.0e-8	1.0e-8	1.0e-8
EMA decay	0.999	0.9999	0.9999	0.9999	0.9999

Table 3: Hyperparameters for the models

D Random samples

In Figures 18, 19 and 20 we show random samples from our production model for some of the prompts from Figure 1.

Figure 18: Random samples from unCLIP for prompt “Vibrant portrait painting of Salvador Dali with a robotic half face”

Figure 19: Random samples from unCLIP for prompt “A close up of a handpalm with leaves growing from it.”

Figure 20: Random samples from unCLIP for prompt “A teddybear on a skateboard in Times Square.”