# Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding

**Chitwan Saharia***, **William Chan***, **Saurabh Saxena**†, **Lala Li**†, **Jay Whang**†, **Emily Denton**, **Seyed Kamyar Seyed Ghasemipour**, **Burcu Karagol Ayan**, **S. Sara Mahdavi**, **Rapha Gontijo Lopes**, **Tim Salimans**, **Jonathan Ho**†, **David J Fleet**†, **Mohammad Norouzi***

{sahariac,williamchan,mnorouzi}@google.com
{srbs,lala,jwhang,jonathanho,davidfleet}@google.com

Google Research, Brain Team
Toronto, Ontario, Canada

## Abstract

We present Imagen, a text-to-image diffusion model with an unprecedented degree of photorealism and a deep level of language understanding. Imagen builds on the power of large transformer language models in understanding text and hinges on the strength of diffusion models in high-fidelity image generation. Our key discovery is that generic large language models (e.g. T5), pretrained on text-only corpora, are surprisingly effective at encoding text for image synthesis: increasing the size of the language model in Imagen boosts both sample fidelity and image-text alignment much more than increasing the size of the image diffusion model. Imagen achieves a new state-of-the-art FID score of 7.27 on the COCO dataset, without ever training on COCO, and human raters find Imagen samples to be on par with the COCO data itself in image-text alignment. To assess text-to-image models in greater depth, we introduce DrawBench, a comprehensive and challenging benchmark for text-to-image models. With DrawBench, we compare Imagen with recent methods including VQ-GAN+CLIP, Latent Diffusion Models, GLIDE and DALL-E 2, and find that human raters prefer Imagen over other models in side-by-side comparisons, both in terms of sample quality and image-text alignment. See imagen.research.google for an overview of the results.

## 1 Introduction

Multimodal learning has come into prominence recently, with text-to-image synthesis [53, 12, 57] and image-text contrastive learning [49, 31, 74] at the forefront. These models have transformed the research community and captured widespread public attention with creative image generation [22, 54] and editing applications [21, 41, 34]. To pursue this research direction further, we introduce Imagen, a text-to-image diffusion model that combines the power of transformer language models (LMs) [15, 52] with high-fidelity diffusion models [28, 29, 16, 41] to deliver an unprecedented degree of photorealism and a deep level of language understanding in text-to-image synthesis. In contrast to prior work that uses only image-text data for model training [e.g., 53, 41], the key finding behind Imagen is that text embeddings from large LMs [52, 15], pretrained on text-only corpora, are remarkably effective for text-to-image synthesis. See Fig. 1 for select samples.

Imagen comprises a frozen T5-XXL [52] encoder to map input text into a sequence of embeddings and a $64\times64$ image diffusion model, followed by two super-resolution diffusion models for generating

*Equal contribution.
†Core contribution.


Sprouts in the shape of text ‘Imagen’ coming out of a fairytale book.


A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.


A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.


Teddy bears swimming at the Olympics 400m Butterfly event.


A cute corgi lives in a house made out of sushi.


A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.


A brain riding a rocketship heading towards the moon.


A dragon fruit wearing karate belt in the snow.


A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.

Figure 1: Select $1024 \times 1024$ Imagen samples for various text inputs. We only include photorealistic images in this figure and leave artistic content to the Appendix, since generating photorealistic images is more challenging from a technical point of view. Figs. A.1 to A.3 show more samples.

$256 \times 256$ and $1024 \times 1024$ images (see Fig. A.4). All diffusion models are conditioned on the text embedding sequence and use classifier-free guidance [27]. Imagen relies on new sampling techniques to allow usage of large guidance weights without sample quality degradation observed in prior work, resulting in images with higher fidelity and better image-text alignment than previously possible.

While conceptually simple and easy to train, Imagen yields surprisingly strong results. Imagen outperforms other methods on COCO [36] with zero-shot FID-30K of 7.27, significantly outperforming prior work such as GLIDE [41] (at 12.4) and the concurrent work of DALL-E 2 [54] (at 10.4). Our zero-shot FID score is also better than state-of-the-art models trained on COCO, e.g., Make-A-Scene [22] (at 7.6). Additionally, human raters indicate that generated samples from Imagen are on-par in image-text alignment to the reference images on COCO captions.

We introduce DrawBench, a new structured suite of text prompts for text-to-image evaluation. DrawBench enables deeper insights through a multi-dimensional evaluation of text-to-image models, with text prompts designed to probe different semantic properties of models. These include compositionality, cardinality, spatial relations, the ability to handle complex text prompts or prompts with rare words, and they include creative prompts that push the limits of models' ability to generate highly implausible scenes well beyond the scope of the training data. With DrawBench, extensive human evaluation shows that Imagen outperforms other recent methods [57, 12, 54] by a significant margin. We further demonstrate some of the clear advantages of the use of large pre-trained language models [52] over multi-modal embeddings such as CLIP [49] as a text encoder for Imagen.

Key contributions of the paper include:

1. We discover that large frozen language models trained only on text data are surprisingly very effective text encoders for text-to-image generation, and that scaling the size of frozen text encoder improves sample quality significantly more than scaling the size of image diffusion model.
2. We introduce *dynamic thresholding*, a new diffusion sampling technique to leverage high guidance weights and generating more photorealistic and detailed images than previously possible.
3. We highlight several important diffusion architecture design choices and propose *Efficient U-Net*, a new architecture variant which is simpler, converges faster and is more memory efficient.
4. We achieve a new state-of-the-art COCO FID of 7.27. Human raters find Imagen to be on-par with the reference images in terms of image-text alignment.
5. We introduce DrawBench, a new comprehensive and challenging evaluation benchmark for the text-to-image task. On DrawBench human evaluation, we find Imagen to outperform all other work, including the concurrent work of DALL-E 2 [54].

## 2 Imagen

Imagen consists of a text encoder that maps text to a sequence of embeddings and a cascade of conditional diffusion models that map these embeddings to images of increasing resolutions (see Fig. A.4). In the following subsections, we describe each of these components in detail.

### 2.1 Pretrained text encoders

Text-to-image models need powerful semantic text encoders to capture the complexity and compositionality of arbitrary natural language text inputs. Text encoders trained on paired image-text data are standard in current text-to-image models; they can be trained from scratch [41, 53] or pretrained on image-text data [54] (e.g., CLIP [49]). The image-text training objectives suggest that these text encoders may encode visually semantic and meaningful representations especially relevant for the text-to-image generation task. Large language models can be another models of choice to encode text for text-to-image generation. Recent progress in large language models (e.g., BERT [15], GPT [47, 48, 7], T5 [52]) have led to leaps in textual understanding and generative capabilities. Language models are trained on text only corpus significantly larger than paired image-text data, thus being exposed to a very rich and wide distribution of text. These models are also generally much larger than text encoders in current image-text models [49, 31, 80] (e.g. PaLM [11] has 540B parameters, while CoCa [80] has a $\approx$ 1B parameter text encoder).

It thus becomes natural to explore both families of text encoders for the text-to-image task. Imagen explores pretrained text encoders: BERT [15], T5 [51] and CLIP [46]. For simplicity, we freeze the weights of these text encoders. Freezing has several advantages such as offline computation of embeddings, resulting in negligible computation or memory footprint during training of the text-to-image model. In our work, we find that there is a clear conviction that scaling the text encoder size improves the quality of text-to-image generation. We also find that while T5-XXL and CLIP

text encoders perform similarly on simple benchmarks such as MS-COCO, human evaluators prefer T5-XXL encoders over CLIP text encoders in both image-text alignment and image fidelity on DrawBench, a set of challenging and compositional prompts. We refer the reader to Section 4.4 for summary of our findings, and Appendix D.1 for detailed ablations.

### 2.2 Diffusion models and classifier-free guidance

Here we give a brief introduction to diffusion models; a precise description is in Appendix A. Diffusion models [63, 28, 65] are a class of generative models that convert Gaussian noise into samples from a learned data distribution via an iterative denoising process. These models can be conditional, for example on class labels, text, or low-resolution images [e.g. 16, 29, 59, 58, 75, 41, 54]. A diffusion model $\hat{\mathbf{x}}_\theta$ is trained on a denoising objective of the form

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},t}\left[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x}\|_2^2\right] \tag{1}$$

where $(\mathbf{x}, \mathbf{c})$ are data-conditioning pairs, $t \sim \mathcal{U}([0, 1])$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\alpha_t, \sigma_t, w_t$ are functions of $t$ that influence sample quality. Intuitively, $\hat{\mathbf{x}}_\theta$ is trained to denoise $\mathbf{z}_t \coloneqq \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$ into $\mathbf{x}$ using a squared error loss, weighted to emphasize certain values of $t$. Sampling such as the ancestral sampler [28] and DDIM [64] start from pure noise $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively generate points $\mathbf{z}_{t_1}, \ldots, \mathbf{z}_{t_T}$, where $1 = t_1 > \cdots > t_T = 0$, that gradually decrease in noise content. These points are functions of the $\mathbf{x}$-predictions $\hat{\mathbf{x}}_0^t \coloneqq \hat{\mathbf{x}}_\theta(\mathbf{z}_t, \mathbf{c})$.

Classifier guidance [16] is a technique to improve sample quality while reducing diversity in conditional diffusion models using gradients from a pretrained model $p(\mathbf{c}|\mathbf{z}_t)$ during sampling. *Classifier-free guidance* [27] is an alternative technique that avoids this pretrained model by instead jointly training a single diffusion model on conditional and unconditional objectives via randomly dropping $\mathbf{c}$ during training (e.g. with 10% probability). Sampling is performed using the adjusted $\mathbf{x}$-prediction $(\mathbf{z}_t - \sigma\tilde{\boldsymbol{\epsilon}}_\theta)/\alpha_t$, where

$$\tilde{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, \mathbf{c}) = w\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}) + (1 - w)\boldsymbol{\epsilon}_\theta(\mathbf{z}_t). \tag{2}$$

Here, $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c})$ and $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t)$ are conditional and unconditional $\boldsymbol{\epsilon}$-predictions, given by $\boldsymbol{\epsilon}_\theta \coloneqq (\mathbf{z}_t - \alpha_t\hat{\mathbf{x}}_\theta)/\sigma_t$, and $w$ is the *guidance weight*. Setting $w = 1$ disables classifier-free guidance, while increasing $w > 1$ strengthens the effect of guidance. Imagen depends critically on classifier-free guidance for effective text conditioning.

### 2.3 Large guidance weight samplers

We corroborate the results of recent text-guided diffusion work [16, 41, 54] and find that increasing the classifier-free guidance weight improves image-text alignment, but damages image fidelity producing highly saturated and unnatural images [27]. We find that this is due to a train-test mismatch arising from high guidance weights. At each sampling step $t$, the $\mathbf{x}$-prediction $\hat{\mathbf{x}}_0^t$ must be within the same bounds as training data $\mathbf{x}$, i.e. within $[-1, 1]$, but we find empirically that high guidance weights cause $\mathbf{x}$-predictions to exceed these bounds. This is a train-test mismatch, and since the diffusion model is iteratively applied on its own output throughout sampling, the sampling process produces unnatural images and sometimes even diverges. To counter this problem, we investigate *static thresholding* and *dynamic thresholding*. See Appendix Fig. A.31 for reference implementation of the techniques and Appendix Fig. A.9 for visualizations of their effects.

**Static thresholding**: We refer to elementwise clipping the $\mathbf{x}$-prediction to $[-1, 1]$ as *static thresholding*. This method was in fact used but not emphasized in previous work [28], and to our knowledge its importance has not been investigated in the context of guided sampling. We discover that static thresholding is essential to sampling with large guidance weights and prevents generation of blank images. Nonetheless, static thresholding still results in over-saturated and less detailed images as the guidance weight further increases.

**Dynamic thresholding**: We introduce a new *dynamic thresholding* method: at each sampling step we set $s$ to a certain percentile absolute pixel value in $\hat{\mathbf{x}}_0^t$, and if $s > 1$, then we threshold $\hat{\mathbf{x}}_0^t$ to the range $[-s, s]$ and then divide by $s$. Dynamic thresholding pushes saturated pixels (those near -1 and 1) inwards, thereby actively preventing pixels from saturation at each step. We find that dynamic thresholding results in significantly better photorealism as well as better image-text alignment, especially when using very large guidance weights.

### 2.4 Robust cascaded diffusion models

Imagen utilizes a pipeline of a base $64 \times 64$ model, and two text-conditional super-resolution diffusion models to upsample a $64 \times 64$ generated image into a $256 \times 256$ image, and then to $1024 \times 1024$ image. Cascaded diffusion models with noise conditioning augmentation [29] have been extremely effective in progressively generating high-fidelity images. Furthermore, making the super-resolution models aware of the amount of noise added, via noise level conditioning, significantly improves the sample quality and helps improving the robustness of the super-resolution models to handle artifacts generated by lower resolution models [29]. Imagen uses noise conditioning augmentation for both the super-resolution models. We find this to be a critical for generating high fidelity images.

Given a conditioning low-resolution image and augmentation level (a.k.a $\mathrm{aug\_level}$) (e.g., strength of Gaussian noise or blur), we corrupt the low-resolution image with the augmentation (corresponding to $\mathrm{aug\_level}$), and condition the diffusion model on $\mathrm{aug\_level}$. During training, $\mathrm{aug\_level}$ is chosen randomly, while during inference, we sweep over its different values to find the best sample quality. In our case, we use Gaussian noise as a form of augmentation, and apply variance preserving Gaussian noise augmentation resembling the forward process used in diffusion models (Appendix A). The augmentation level is specified using $\mathrm{aug\_level} \in [0, 1]$. See Fig. A.32 for reference pseudocode.

### 2.5 Neural network architecture

**Base model**: We adapt the U-Net architecture from [40] for our base $64 \times 64$ text-to-image diffusion model. The network is conditioned on text embeddings via a pooled embedding vector, added to the diffusion timestep embedding similar to the class embedding conditioning method used in [16, 29]. We further condition on the entire sequence of text embeddings by adding cross attention [57] over the text embeddings at multiple resolutions. We study various methods of text conditioning in Appendix D.3.1. Furthermore, we found Layer Normalization [2] for text embeddings in the attention and pooling layers to help considerably improve performance.

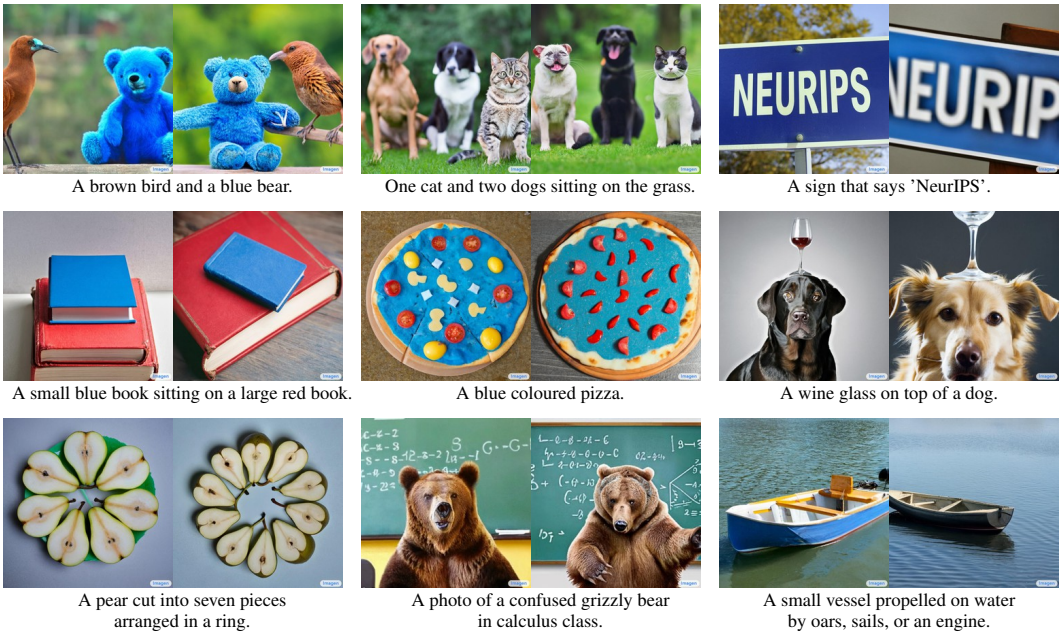
**Super-resolution models**: For $64 \times 64 \rightarrow 256 \times 256$ super-resolution, we use the U-Net model adapted from [40, 58]. We make several modifications to this U-Net model for improving memory efficiency, inference time and convergence speed (our variant is 2-3x faster in steps/second over the U-Net used in [40, 58]). We call this variant *Efficient U-Net* (See Appendix B.1 for more details and comparisons). Our $256 \times 256 \rightarrow 1024 \times 1024$ super-resolution model trains on $64 \times 64 \rightarrow 256 \times 256$ crops of the $1024 \times 1024$ image. To facilitate this, we remove the self-attention layers, however we keep the text cross-attention layers which we found to be critical. During inference, the model receives the full $256 \times 256$ low-resolution images as inputs, and returns upsampled $1024 \times 1024$ images as outputs. Note that we use text cross attention for both our super-resolution models.

## 3 Evaluating Text-to-Image Models

The COCO [36] validation set is the standard benchmark for evaluating text-to-image models for both the supervised [82, 22] and the zero-shot setting [53, 41]. The key automated performance metrics used are FID [26] to measure image fidelity, and CLIP score [25, 49] to measure image-text alignment. Consistent with previous works, we report zero-shot FID-30K, for which 30K prompts are drawn randomly from the validation set, and the model samples generated on these prompts are compared with reference images from the full validation set. Since guidance weight is an important ingredient to control image quality and text alignment, we report most of our ablation results using trade-off (or *pareto*) curves between CLIP and FID scores across a range of guidance weights.

Both FID and CLIP scores have limitations, for example FID is not fully aligned with perceptual quality [42], and CLIP is ineffective at counting [49]. Due to these limitations, we use human evaluation to assess image quality and caption similarity, with ground truth reference caption-image pairs as a baseline. We use two experimental paradigms:

1. To probe image quality, the rater is asked to select between the model generation and reference image using the question: "Which image is more photorealistic (looks more real)?". We report the percentage of times raters choose model generations over reference images (the *preference rate*).
2. To probe alignment, human raters are shown an image and a prompt and asked "Does the caption accurately describe the above image?". They must respond with "yes", "somewhat", or "no". These responses are scored as 100, 50, and 0, respectively. These ratings are obtained independently for model samples and reference images, and both are reported.




A brown bird and a blue bear. One cat and two dogs sitting on the grass. A sign that says 'NeurIPS'.

A small blue book sitting on a large red book. A blue coloured pizza. A wine glass on top of a dog.

A pear cut into seven pieces arranged in a ring. A photo of a confused grizzly bear in calculus class. A small vessel propelled on water by oars, sails, or an engine.

Figure 2: Non-cherry picked Imagen samples for different categories of prompts from DrawBench.

For both cases we use 200 randomly chosen image-caption pairs from the COCO validation set. Subjects were shown batches of 50 images. We also used interleaved "control" trials, and only include rater data from those who correctly answered at least 80% of the control questions. This netted 73 and 51 ratings per image for image quality and image-text alignment evaluations, respectively.

**DrawBench**: While COCO is a valuable benchmark, it is increasingly clear that it has a limited spectrum of prompts that do not readily provide insight into differences between models (e.g., see Sec. 4.2). Recent work by [10] proposed a new evaluation set called PaintSkills to systematically evaluate visual reasoning skills and social biases beyond COCO. With similar motivation, we introduce *DrawBench*, a comprehensive and challenging set of prompts that support the evaluation and comparison of text-to-image models. DrawBench contains 11 categories of prompts, testing different capabilities of models such as the ability to faithfully render different colors, numbers of objects, spatial relations, text in the scene, and unusual interactions between objects. Categories also include complex prompts, including long, intricate textual descriptions, rare words, and also misspelled prompts. We also include sets of prompts collected from DALL-E [53], Gary Marcus et al. [38] and Reddit. Across these 11 categories, DrawBench comprises 200 prompts in total, striking a good balance between the desire for a large, comprehensive dataset, and small enough that human evaluation remains feasible. (Appendix C provides a more detailed description of DrawBench. Fig. 2 shows example prompts from DrawBench with Imagen samples.)

We use DrawBench to directly compare different models. To this end, human raters are presented with two sets of images, one from Model A and one from Model B, each of which has 8 samples. Human raters are asked to compare Model A and Model B on sample fidelity and image-text alignment. They respond with one of three choices: Prefer Model A; Indifferent; or Prefer Model B.

## 4 Experiments

Section 4.1 describes training details, Sections 4.2 and 4.3 analyze results on MS-COCO and DrawBench, and Section 4.4 summarizes our ablation studies and key findings. For all experiments below, the images are fair random samples from Imagen with no post-processing or re-ranking.

### 4.1 Training details

Unless specified, we train a 2B parameter model for the $64 \times 64$ text-to-image synthesis, and 600M and 400M parameter models for $64 \times 64 \rightarrow 256 \times 256$ and $256 \times 256 \rightarrow 1024 \times 1024$ for super-resolution respectively. We use a batch size of 2048 and 2.5M training steps for all models. We use 256 TPU-v4 chips for our base $64 \times 64$ model, and 128 TPU-v4 chips for both super-resolution

Table 1: MS-COCO $256 \times 256$ FID-30K. We use a guidance weight of 1.35 for our $64 \times 64$ model, and a guidance weight of 8.0 for our super-resolution model.

| Model | FID-30K | Zero-shot FID-30K |
|---|---|---|
| AttnGAN [76] | 35.49 | |
| DM-GAN [83] | 32.64 | |
| DF-GAN [69] | 21.42 | |
| DM-GAN + CL [78] | 20.79 | |
| XMC-GAN [81] | 9.33 | |
| LAFITE [82] | 8.12 | |
| Make-A-Scene [22] | 7.55 | |
| DALL-E [53] | | 17.89 |
| LAFITE [82] | | 26.94 |
| GLIDE [41] | | 12.24 |
| DALL-E 2 [54] | | 10.39 |
| **Imagen (Our Work)** | | **7.27** |

Table 2: COCO $256 \times 256$ human evaluation comparing model outputs and original images. For the bottom part (no people), we filter out prompts containing one of `man`, `men`, `woman`, `women`, `person`, `people`, `child`, `adult`, `adults`, `boy`, `boys`, `girl`, `girls`, `guy`, `lady`, `ladies`, `someone`, `toddler`, `(sport) player`, `workers`, `spectators`.

| Model | Photorealism ↑ | Alignment ↑ |
|---|---|---|
| *Original* | | |
| Original | 50.0% | $91.9 \pm 0.42$ |
| Imagen | $39.5 \pm 0.75\%$ | $91.4 \pm 0.44$ |
| *No people* | | |
| Original | 50.0% | $92.2 \pm 0.54$ |
| Imagen | $43.9 \pm 1.01\%$ | $92.1 \pm 0.55$ |

models. We do not find over-fitting to be an issue, and we believe further training might improve overall performance. We use Adafactor for our base $64 \times 64$ model, because initial comparisons with Adam suggested similar performance with much smaller memory footprint for Adafactor. For super-resolution models, we use Adam as we found Adafactor to hurt model quality in our initial ablations. For classifier-free guidance, we joint-train unconditionally via zeroing out the text embeddings with 10% probability for all three models. We train on a combination of internal datasets, with $\approx$ 460M image-text pairs, and the publicly available Laion dataset [61], with $\approx$ 400M image-text pairs. There are limitations in our training data, and we refer the reader to Section 6 for details. See Appendix F for more implementation details.

### 4.2 Results on COCO

We evaluate Imagen on the COCO validation set using FID score, similar to [53, 41]. Table 1 displays the results. Imagen achieves state of the art *zero-shot* FID on COCO at 7.27, outperforming the concurrent work of DALL-E 2 [54] and even models trained on COCO. Table 2 reports the human evaluation to test image quality and alignment on the COCO validation set. We report results on the original COCO validation set, as well as a filtered version in which all reference data with people have been removed. For photorealism, Imagen achieves 39.2% preference rate indicating high image quality generation. On the set with no people, there is a boost in preference rate of Imagen to 43.6%, indicating Imagen's limited ability to generate photorealistic people. On caption similarity, Imagen's score is on-par with the original reference images, suggesting Imagen's ability to generate images that align well with COCO captions.

### 4.3 Results on DrawBench

Using DrawBench, we compare Imagen with DALL-E 2 (the public version) [54], GLIDE [41], Latent Diffusion [57], and CLIP-guided VQ-GAN [12]. Fig. 3 shows the human evaluation results for pairwise comparison of Imagen with each of the three models. We report the percentage of time raters prefer Model A, Model B, or are indifferent for both image fidelity and image-text alignment. We aggregate the scores across all the categories and raters. We find the human raters to exceedingly prefer Imagen over all others models in both image-text alignment and image fidelity. We refer the reader to Appendix E for a more detailed category wise comparison and qualitative comparison.

### 4.4 Analysis of Imagen

For a detailed analysis of Imagen see Appendix D. Key findings are discussed in Fig. 4 and below.

**Scaling text encoder size is extremely effective.** We observe that scaling the size of the text encoder leads to consistent improvement in both image-text alignment and image fidelity. Imagen trained with our largest text encoder, T5-XXL (4.6B parameters), yields the best results (Fig. 4a).

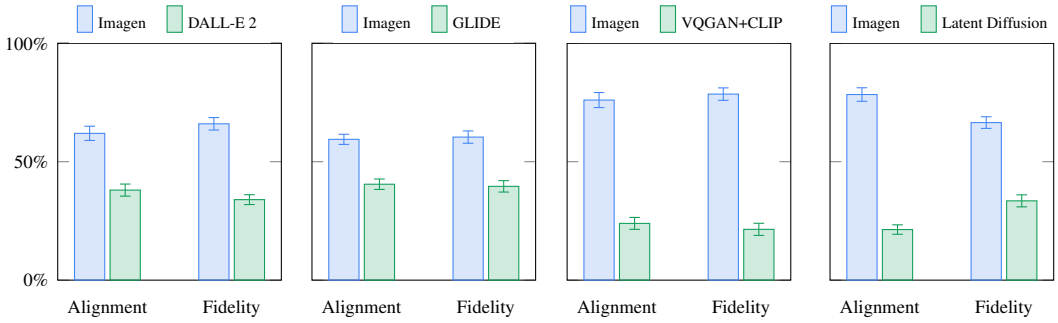
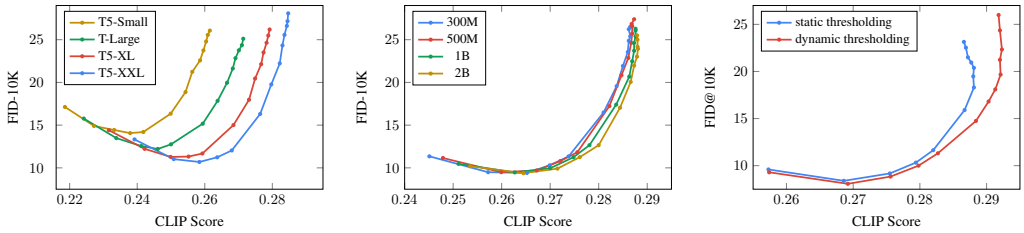



Figure 3: Comparison between Imagen and DALL-E 2 [54], GLIDE [41], VQ-GAN+CLIP [12] and Latent Diffusion [57] on DrawBench: User preference rates (with 95% confidence intervals) for image-text alignment and image fidelity.




(a) Impact of encoder size. (b) Impact of U-Net size. (c) Impact of thresholding.

Figure 4: Summary of some of the critical findings of Imagen with pareto curves sweeping over different guidance values. See Appendix D for more details.

**Scaling text encoder size is more important than U-Net size.** While scaling the size of the diffusion model U-Net improves sample quality, we found scaling the text encoder size to be significantly more impactful than the U-Net size (Fig. 4b).

**Dynamic thresholding is critical.** We show that dynamic thresholding results in samples with significantly better photorealism and alignment with text, over static or no thresholding, especially under the presence of large classifier-free guidance weights (Fig. 4c).

**Human raters prefer T5-XXL over CLIP on DrawBench.** The models trained with T5-XXL and CLIP text encoders perform similarly on the COCO validation set in terms of CLIP and FID scores. However, we find that human raters prefer T5-XXL over CLIP on DrawBench across all 11 categories.

**Noise conditioning augmentation is critical.** We show that training the super-resolution models with noise conditioning augmentation leads to better CLIP and FID scores. We also show that noise conditioning augmentation enables stronger text conditioning for the super-resolution model, resulting in improved CLIP and FID scores at higher guidance weights. Adding noise to the low-res image during inference along with the use of large guidance weights allows the super-resolution models to generate diverse upsampled outputs while removing artifacts from the low-res image.

**Text conditioning method is critical.** We observe that conditioning over the sequence of text embeddings with cross attention significantly outperforms simple mean or attention based pooling in both sample fidelity as well as image-text alignment.

**Efficient U-Net is critical.** Our Efficient U-Net implementation uses less memory, converges faster, and has better sample quality with faster inference.

## 5 Related Work

Diffusion models have seen wide success in image generation [28, 40, 59, 16, 29, 58], outperforming GANs in fidelity and diversity, without training instability and mode collapse issues [6, 16, 29]. Autoregressive models [37], GANs [76, 81], VQ-VAE Transformer-based methods [53, 22], and diffusion models have seen remarkable progress in text-to-image [57, 41, 57], including the concurrent DALL-E 2 [54], which uses a diffusion prior on CLIP text latents and cascaded diffusion models

to generate high resolution $1024 \times 1024$ images; we believe Imagen is much simpler, as Imagen does not need to learn a latent prior, yet achieves better results in both MS-COCO FID and human evaluation on DrawBench. GLIDE [41] also uses cascaded diffusion models for text-to-image, but we use large pretrained frozen language models, which we found to be instrumental to both image fidelity and image-text alignment. XMC-GAN [81] also uses BERT as a text encoder, but we scale to much larger text encoders and demonstrate the effectiveness thereof. The use of cascaded models is also popular throughout the literature [14, 39] and has been used with success in diffusion models to generate high resolution images [16, 29].

## 6 Conclusions, Limitations and Societal Impact

Imagen showcases the effectiveness of frozen large pretrained language models as text encoders for the text-to-image generation using diffusion models. Our observation that scaling the size of these language models have significantly more impact than scaling the U-Net size on overall performance encourages future research directions on exploring even bigger language models as text encoders. Furthermore, through Imagen we re-emphasize the importance of classifier-free guidance, and we introduce dynamic thresholding, which allows usage of much higher guidance weights than seen in previous works. With these novel components, Imagen produces $1024 \times 1024$ samples with unprecedented photorealism and alignment with text.

Our primary aim with Imagen is to advance research on generative methods, using text-to-image synthesis as a test bed. While end-user applications of generative methods remain largely out of scope, we recognize the potential downstream applications of this research are varied and may impact society in complex ways. On the one hand, generative models have a great potential to complement, extend, and augment human creativity [30]. Text-to-image generation models, in particular, have the potential to extend image-editing capabilities and lead to the development of new tools for creative practitioners. On the other hand, generative methods can be leveraged for malicious purposes, including harassment and misinformation spread [20], and raise many concerns regarding social and cultural exclusion and bias [67, 62, 68]. These considerations inform our decision to not to release code or a public demo. In future work we will explore a framework for responsible externalization that balances the value of external auditing with the risks of unrestricted open-access.

Another ethical challenge relates to the large scale data requirements of text-to-image models, which have have led researchers to rely heavily on large, mostly uncurated, web-scraped datasets. While this approach has enabled rapid algorithmic advances in recent years, datasets of this nature have been critiqued and contested along various ethical dimensions. For example, public and academic discourse regarding appropriate use of public data has raised concerns regarding data subject awareness and consent [24, 18, 60, 43]. Dataset audits have revealed these datasets tend to reflect social stereotypes, oppressive viewpoints, and derogatory, or otherwise harmful, associations to marginalized identity groups [44, 4]. Training text-to-image models on this data risks reproducing these associations and causing significant representational harm that would disproportionately impact individuals and communities already experiencing marginalization, discrimination and exclusion within society. As such, there are a multitude of data challenges that must be addressed before text-to-image models like Imagen can be safely integrated into user-facing applications. While we do not directly address these challenges in this work, an awareness of the limitations of our training data guide our decision not to release Imagen for public use. We strongly caution against the use text-to-image generation methods for any user-facing tools without close care and attention to the contents of the training dataset.

Imagen's training data was drawn from several pre-existing datasets of image and English alt-text pairs. A subset of this data was filtered to removed noise and undesirable content, such as pornographic imagery and toxic language. However, a recent audit of one of our data sources, LAION-400M [61], uncovered a wide range of inappropriate content including pornographic imagery, racist slurs, and harmful social stereotypes [4]. This finding informs our assessment that Imagen is not suitable for public use at this time and also demonstrates the value of rigorous dataset audits and comprehensive dataset documentation (e.g. [23, 45]) in informing consequent decisions about the model's appropriate and safe use. Imagen also relies on text encoders trained on uncurated web-scale data, and thus inherits the social biases and limitations of large language models [5, 3, 50].

While we leave an in-depth empirical analysis of social and cultural biases encoded by Imagen to future work, our small scale internal assessments reveal several limitations that guide our decision not to release Imagen at this time. First, all generative models, including Imagen, Imagen, may run into danger of dropping modes of the data distribution, which may further compound the social

consequence of dataset bias. Second, Imagen exhibits serious limitations when generating images depicting people. Our human evaluations found Imagen obtains significantly higher preference rates when evaluated on images that do not portray people, indicating a degradation in image fidelity. Finally, our preliminary assessment also suggests Imagen encodes several social biases and stereotypes, including an overall bias towards generating images of people with lighter skin tones and a tendency for images portraying different professions to align with Western gender stereotypes. Even when we focus generations away from people, our preliminary analysis indicates Imagen encodes a range of social and cultural biases when generating images of activities, events, and objects.

While there has been extensive work auditing image-to-text and image labeling models for forms of social bias (e.g. [8, 9, 68]), there has been comparatively less work on social bias evaluation methods for text-to-image models, with the recent exception of [10]. We believe this is a critical avenue for future research and we intend to explore benchmark evaluations for social and cultural bias in future work—for example, exploring whether it is possible to generalize the normalized pointwise mutual information metric [1] to the measurement of biases in image generation models. There is also a great need to develop a conceptual vocabulary around potential harms of text-to-image models that could guide the development of evaluation metrics and inform responsible model release. We aim to address these challenges in future work.

## 7 Acknowledgements

We give thanks to Ben Poole for reviewing our manuscript, early discussions, and providing many helpful comments and suggestions throughout the project. Special thanks to Kathy Meier-Hellstern, Austin Tarango, and Sarah Laszlo for helping us incorporate important responsible AI practices around this project. We appreciate valuable feedback and support from Elizabeth Adkison, Zoubin Ghahramani, Jeff Dean, Yonghui Wu, and Eli Collins. We are grateful to Tom Small for designing the Imagen watermark. We thank Jason Baldridge, Han Zhang, and Kevin Murphy for initial discussions and feedback. We acknowledge hard work and support from Fred Alcober, Hibaq Ali, Marian Croak, Aaron Donsbach, Tulsee Doshi, Toju Duke, Douglas Eck, Jason Freidenfelds, Brian Gabriel, Molly FitzMorris, David Ha, Philip Parham, Laura Pearce, Evan Rapoport, Lauren Skelly, Johnny Soraker, Negar Rostamzadeh, Vijay Vasudevan, Tris Warkentin, Jeremy Weinstein, and Hugh Williams for giving us advice along the project and assisting us with the publication process. We thank Victor Gomes and Erica Moreira for their consistent and critical help with TPU resource allocation. We also give thanks to Shekoofeh Azizi, Harris Chan, Chris A. Lee, and Nick Ma for volunteering a considerable amount of their time for testing out DrawBench. We thank Aditya Ramesh, Prafulla Dhariwal, and Alex Nichol for allowing us to use DALL-E 2 samples and providing us with GLIDE samples. We are thankful to Matthew Johnson and Roy Frostig for starting the JAX project and to the whole JAX team for building such a fantastic system for high-performance machine learning research. Special thanks to Durk Kingma, Jascha Sohl-Dickstein, Lucas Theis and the Toronto Brain team for helpful discussions and spending time Imagening!

## References

[1] Osman Aka, Ken Burke, Alex Bauerle, Christina Greer, and Margaret Mitchell. Measuring Model Biases in the Absence of Ground Truth. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? 🦜. In *Proceedings of FAccT 2021*, 2021.

[4] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. In *arXiv:2110.01963*, 2021.

[5] Shikha Bordia and Samuel R. Bowman. Identifying and Reducing Gender Bias in Word-Level Language Models. In *NAACL*, 2017.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[7] 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. In *NeurIPS*, 2020.

[8] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, Proceedings of Machine Learning Research. PMLR, 2018.

[9] Kaylee Burns, Lisa Hendricks, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *European Conference on Computer Vision (ECCV)*, 2018.

[10] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generative transformers. *arxiv:2202.04053*, 2022.

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling Language Modeling with Pathways. In *arXiv:2001.08361*, 2022.

[12] Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. Vqgan-clip: Open domain image generation and editing with natural language guidance. *arXiv preprint arXiv:2204.08583*, 2022.

[13] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34, 2021.

[14] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *NIPS*, 2015.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, 2019.

[16] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2022.

[17] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Dulhanty, Chris. Issues in Computer Vision Data Collection: Bias, Consent, and Label Taxonomy. In *UWSpace*, 2020.

[19] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12873–12883, 2021.

[20] Mary Anne Franks and Ari Ezra Waldman. Sex, lies and videotape: deep fakes and free speech delusions. *Maryland Law Review*, 78(4):892–898, 2019.

[21] Tsu-Jui Fu, Xin Eric Wang, and William Yang Wang. Language-Driven Image Style Transfer. *arXiv preprint arXiv:2106.00178*, 2021.

[22] 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 preprint arXiv:2203.13131*, 2022.

[23] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for Datasets. *arXiv:1803.09010 [cs]*, March 2020.

[24] Adam Harvey and Jules LaPlace. MegaPixels: Origins and endpoints of biometric datasets "In the Wild". https://megapixels.cc, 2019.

[25] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

[26] 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. *arXiv preprint arXiv:1706.08500*, 2017.

[27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *NeurIPS*, 2020.

[29] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 2022.

[30] Rowan T. Hughes, Liming Zhu, and Tomasz Bednarz. Generative adversarial networks-enabled human-artificial intelligence collaborative applications for creative and design industries: A systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 4, 2021.

[31] 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. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[32] Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

[33] Zahra Kadkhodaie and Eero P Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34, 2021.

[34] Gwanghyun Kim and Jong Chul Ye. Diffusionclip: Text-guided image manipulation using diffusion models. *arXiv preprint arXiv:2110.02711*, 2021.

[35] Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

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

[37] Elman Mansimov, Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Generating Images from Captions with Attention. In *ICLR*, 2016.

[38] Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. In *arXiv:2204.13807*, 2022.

[39] Jacob Menick and Nal Kalchbrenner. Generating High Fidelity Images with Subscale Pixel Networks and Multidimensional Upscaling. In *ICLR*, 2019.

[40] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

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

[42] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On Aliased Resizing and Surprising Subtleties in GAN Evaluation. In *CVPR*, 2022.

[43] Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336, 2021.

[44] Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision? *arXiv:2006.16923*, 2020.

[45] Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data cards: Purposeful and transparent dataset documentation for responsible ai. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[46] Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to Generate Reviews and Discovering Sentiment. In *arXiv:1704.01444*, 2017.

[47] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. In *preprint*, 2018.

[48] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. In *preprint*, 2019.

[49] 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. In *ICML*, 2021.

[50] Jack Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George Driessche, Lisa Hendricks, Maribeth Rauh, Po-Sen Huang, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training gopher. *arXiv:2112.11446*, 2021.

[51] Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. Online and Linear-Time Attention by Enforcing Monotonic Alignments. In *ICML*, 2017.

[52] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *JMLR*, 21(140), 2020.

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

[54] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. In *arXiv*, 2022.

[55] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[56] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *International conference on machine learning*, pages 1060–1069. PMLR, 2016.

[57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*, 2022.

[58] Chitwan Saharia, William Chan, Huiwen Chang, Chris A. Lee, Jonathan Ho, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Palette: Image-to-Image Diffusion Models. In *arXiv:2111.05826*, 2021.

[59] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021.

[60] Morgan Klaus Scheuerman, Emily L. Denton, and A. Hanna. Do datasets have politics? disciplinary values in computer vision dataset development. *Proceedings of the ACM on Human-Computer Interaction*, 5:1 – 37, 2021.

[61] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[62] Lucas Sequeira, Bruno Moreschi, Amanda Jurno, and Vinicius Arruda dos Santos. Which faces can AI generate? Normativity, whiteness and lack of diversity in This Person Does Not Exist. In *CVPR Workshop Beyond Fairness: Towards a Just, Equitable, and Accountable Computer Vision*, 2021.

[63] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

[64] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[65] Yang Song and Stefano Ermon. Generative Modeling by Estimating Gradients of the Data Distribution. *NeurIPS*, 2019.

[66] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

[67] Ramya Srinivasan and Kanji Uchino. Biases in generative art: A causal look from the lens of art history. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, page 41–51, 2021.

[68] Ryan Steed and Aylin Caliskan. Image representations learned with unsupervised pre-training contain human-like biases. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 701–713. Association for Computing Machinery, 2021.

[69] 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 preprint arXiv:2008.05865*, 2020.

[70] Belinda Tzen and Maxim Raginsky. Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit. In *arXiv:1905.09883*, 2019.

[71] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[72] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.

[73] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[74] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[75] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. *arXiv preprint arXiv:2112.02475*, 2021.

[76] 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. In *CVPR*, 2018.

[77] 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. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324, 2018.

[78] Hui Ye, Xiulong Yang, Martin Takac, Rajshekhar Sunderraman, and Shihao Ji. Improving text-to-image synthesis using contrastive learning. *arXiv preprint arXiv:2107.02423*, 2021.

[79] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

[80] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

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

[82] 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 preprint arXiv:2111.13792*, 2021.

[83] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5802–5810, 2019.

[84] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *ICCV*, 2015.


A chromeplated cat sculpture placed on a Persian rug.

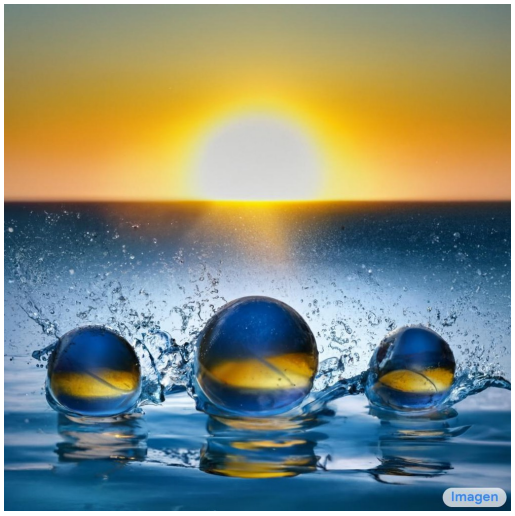

Android Mascot made from bamboo.


Intricate origami of a fox and a unicorn in a snowy forest.


A transparent sculpture of a duck made out of glass.


A raccoon wearing cowboy hat and black leather jacket is behind the backyard window. Rain droplets on the window.


A bucket bag made of blue suede. The bag is decorated with intricate golden paisley patterns. The handle of the bag is made of rubies and pearls.




Three spheres made of glass falling into ocean. Water is splashing. Sun is setting.


Vines in the shape of text 'Imagen' with flowers and butterflies bursting out of an old TV.


A strawberry splashing in the coffee in a mug under the starry sky.

Figure A.1: Select $1024 \times 1024$ Imagen samples for various text inputs.

A wall in a royal castle. There are two paintings on the wall. The one on the left a detailed oil painting of the royal raccoon king. The one on the right a detailed oil painting of the royal raccoon queen.

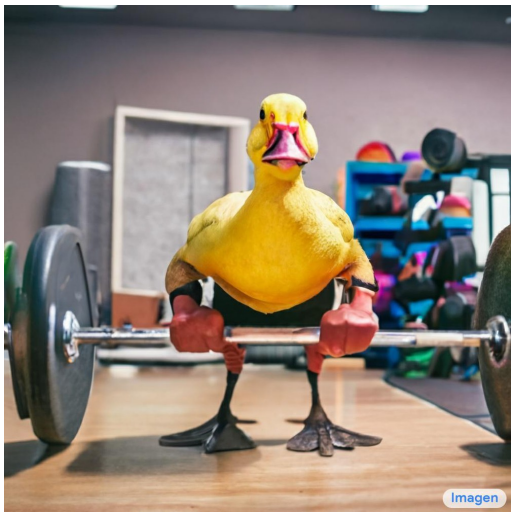
A group of teddy bears in suit in a corporate office celebrating the birthday of their friend. There is a pizza cake on the desk.

A chrome-plated duck with a golden beak arguing with an angry turtle in a forest.

A family of three houses in a meadow. The Dad house is a large blue house. The Mom house is a large pink house. The Child house is a small wooden shed.

A cloud in the shape of two bunnies playing with a ball. The ball is made of clouds too.

A Pomeranian is sitting on the Kings throne wearing a crown. Two tiger soldiers are standing next to the throne.



An angry duck doing heavy weightlifting at the gym.

A dslr picture of colorful graffiti showing a hamster with a moustache.

A photo of a person with the head of a cow, wearing a tuxedo and black bowtie. Beach wallpaper in the background.

Figure A.2: Select $1024 \times 1024$ Imagen samples for various text inputs.

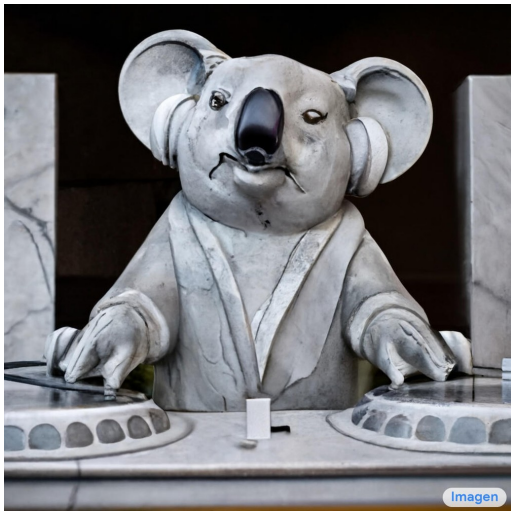

A relaxed garlic with a blindfold reading a newspaper while floating in a pool of tomato soup.


A photo of a corgi dog wearing a wizard hat playing guitar on the top of a mountain.


A single beam of light enter the room from the ceiling. The beam of light is illuminating an easel. On the easel there is a Rembrandt painting of a raccoon.


A squirrel is inside a giant bright shiny crystal ball in on the surface of blue ocean. There are few clouds in the sky.


A bald eagle made of chocolate powder, mango, and whipped cream.




A marble statue of a Koala DJ in front of a marble statue of a turntable. The Koala has wearing large marble headphones.


The Toronto skyline with Google brain logo written in fireworks.


A blue jay standing on a large basket of rainbow macarons.


An art gallery displaying Monet paintings. The art gallery is flooded. Robots are going around the art gallery using paddle boards.

Figure A.3: Select $1024 \times 1024$ Imagen samples for various text inputs.

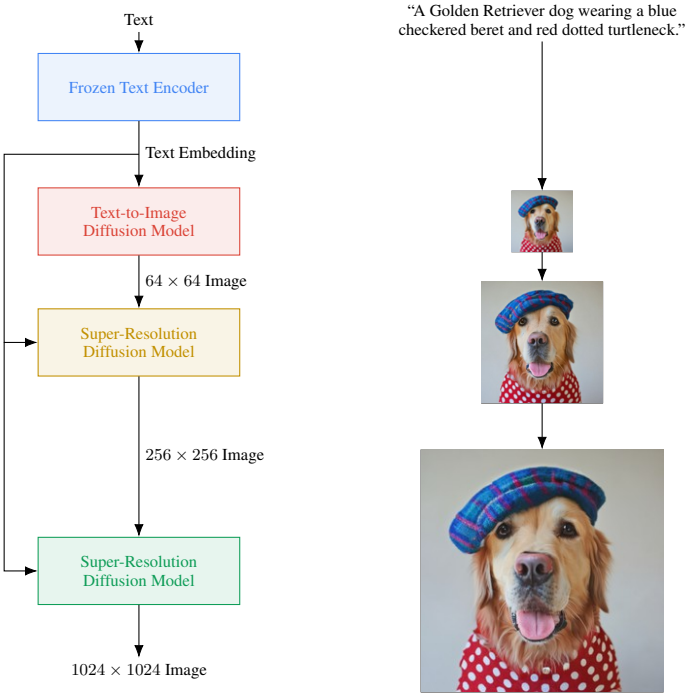



Figure A.4: Visualization of Imagen. Imagen uses a frozen text encoder to encode the input text into text embeddings. A conditional diffusion model maps the text embedding into a $64 \times 64$ image. Imagen further utilizes text-conditional super-resolution diffusion models to upsample the image, first $64 \times 64 \rightarrow 256 \times 256$, and then $256 \times 256 \rightarrow 1024 \times 1024$.

# A Background

Diffusion models are latent variable models with latents $\mathbf{z} = \{\mathbf{z}_t \mid t \in [0, 1]\}$ that obey a *forward process* $q(\mathbf{z}|\mathbf{x})$ starting at data $\mathbf{x} \sim p(\mathbf{x})$. This forward process is a Gaussian process that satisfies the Markovian structure:

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}(\mathbf{z}_t; \alpha_t \mathbf{x}, \sigma_t^2 \mathbf{I}), \quad q(\mathbf{z}_t|\mathbf{z}_s) = \mathcal{N}(\mathbf{z}_t; (\alpha_t/\alpha_s)\mathbf{z}_s, \sigma_{t|s}^2 \mathbf{I}) \tag{3}$$

where $0 \leq s < t \leq 1$, $\sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$, and $\alpha_t, \sigma_t$ specify a differentiable *noise schedule* whose log signal-to-noise-ratio, i.e., $\lambda_t = \log[\alpha_t^2/\sigma_t^2]$, decreases with $t$ until $q(\mathbf{z}_1) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$. For generation, the diffusion model is learned to *reverse* this forward process.

Learning to reverse the forward process can be reduced to learning to denoise $\mathbf{z}_t \sim q(\mathbf{z}_t|\mathbf{x})$ into an estimate $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}) \approx \mathbf{x}$ for all $t$, where $\mathbf{c}$ is an optional conditioning signal (such as text embeddings or a low resolution image) drawn from the dataset jointly with $\mathbf{x}$. This is accomplished training $\hat{\mathbf{x}}_\theta$ using a weighted squared error loss

$$\mathbb{E}_{\boldsymbol{\epsilon}, t}\left[w(\lambda_t)\|\hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}) - \mathbf{x}\|_2^2\right] \tag{4}$$

where $t \sim \mathcal{U}([0, 1])$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}$. This reduction of generation to denoising is justified as optimizing a weighted variational lower bound on the data log likelihood under the diffusion model, or as a form of denoising score matching [72, 65, 28, 35]. We use the $\boldsymbol{\epsilon}$-prediction parameterization, defined as $\hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}) = (\mathbf{z}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}))/\alpha_t$, and we impose a squared error loss on $\boldsymbol{\epsilon}_\theta$ in $\boldsymbol{\epsilon}$ space with $t$ sampled according to a cosine schedule [40]. This corresponds to a particular weighting $w(\lambda_t)$ and leads to a scaled score estimate $\boldsymbol{\epsilon}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}) \approx -\sigma_t \nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t|\mathbf{c})$, where $p(\mathbf{z}_t|\mathbf{c})$ is the true density of $\mathbf{z}_t$ given $\mathbf{c}$ under the forward process starting at $\mathbf{x} \sim p(\mathbf{x})$ [28, 35, 66]. Related model designs include the work of [70, 32, 33].

To sample from the diffusion model, we start at $\mathbf{z}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and use the discrete time ancestral sampler [28] and DDIM [64] for certain models. DDIM follows the deterministic update rule

$$\mathbf{z}_s = \alpha_s \hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c}) + \frac{\sigma_s}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c})) \tag{5}$$

where $s < t$ follow a uniformly spaced sequence from 1 to 0. The ancestral sampler arises from a reversed description of the forward process; noting that $q(\mathbf{z}_s|\mathbf{z}_t, \mathbf{x}) = \mathcal{N}(\mathbf{z}_s; \tilde{\boldsymbol{\mu}}_{s|t}(\mathbf{z}_t, \mathbf{x}), \tilde{\sigma}_{s|t}^2 \mathbf{I})$, where $\tilde{\boldsymbol{\mu}}_{s|t}(\mathbf{z}_t, \mathbf{x}) = e^{\lambda_t - \lambda_s}(\alpha_s/\alpha_t)\mathbf{z}_t + (1 - e^{\lambda_t - \lambda_s})\alpha_s \mathbf{x}$ and $\tilde{\sigma}_{s|t}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_s^2$, it follows the stochastic update rule

$$\mathbf{z}_s = \tilde{\boldsymbol{\mu}}_{s|t}(\mathbf{z}_t, \hat{\mathbf{x}}_\theta(\mathbf{z}_t, \lambda_t, \mathbf{c})) + \sqrt{(\tilde{\sigma}_{s|t}^2)^{1-\gamma}(\sigma_{t|s}^2)^\gamma}\, \boldsymbol{\epsilon} \tag{6}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\gamma$ controls the stochasticity of the sampler [40].

# B Architecture Details

## B.1 Efficient U-Net

We introduce a new architectural variant, which we term Efficient U-Net, for our super-resolution models. We find our Efficient U-Net to be simpler, converges faster, and is more memory efficient compared to some prior implementations [40], especially for high resolutions. We make several key modifications to the U-Net architecture, such as shifting of model parameters from high resolution blocks to low resolution, scaling the skip connections by $1/\sqrt{2}$ similar to [66, 59] and reversing the order of downsampling/upsampling operations in order to improve the speed of the forward pass. Efficient U-Net makes several key modifications to the typical U-Net model used in [16, 58]:

- We shift the model parameters from the high resolution blocks to the low resolution blocks, via adding more residual blocks for the lower resolutions. Since lower resolution blocks typically have many more channels, this allows us to increase the model capacity through more model parameters, without egregious memory and computation costs.
- When using large number of residual blocks at lower-resolution (e.g. we use 8 residual blocks at lower-resolutions compared to typical 2-3 residual blocks used in standard U-Net architectures [16, 59]) we find that scaling the skip connections by $1/\sqrt{2}$ similar to [66, 59] significantly improves convergence speed.

- In a typical U-Net's downsampling block, the downsampling operation happens after the convolutions, and in an upsampling block, the upsampling operation happens prior the convolution. We reverse this order for both downsampling and upsampling blocks in order to significantly improve the speed of the forward pass of the U-Net, and find no performance degradation.

With these key simple modifications, Efficient U-Net is simpler, converges faster, and is more memory efficient compared to some prior U-Net implementations. Fig. A.30 shows the full architecture of Efficient U-Net, while Figures A.28 and A.29 show detailed description of the Downsampling and Upsampling blocks of Efficient U-Net respectively. See Appendix D.3.2 for results.

# C DrawBench

In this section, we describe our new benchmark for fine-grained analysis of text-to-image models, namely, DrawBench. DrawBench consists of 11 categories with approximately 200 text prompts. This is large enough to test the model well, while small enough to easily perform trials with human raters. Table A.1 enumerates these categories along with description and few examples. We release the full set of samples here.

For evaluation on this benchmark, we conduct an independent human evaluation run for each category. For each prompt, the rater is shown two sets of images - one from Model A, and second from Model B. Each set contains 8 random (non-cherry picked) generations from the corresponding model. The rater is asked two questions -

1. Which set of images is of higher quality?
2. Which set of images better represents the text caption : {Text Caption}?

where the questions are designed to measure: 1) image fidelity, and 2) image-text alignment. For each question, the rater is asked to select from three choices:

1. I prefer set A.
2. I am indifferent.
3. I prefer set B.

We aggregate scores from 25 raters for each category (totalling to $25 \times 11 = 275$ raters). We do not perform any post filtering of the data to identify unreliable raters, both for expedience and because the task was straightforward to explain and execute.

# D Imagen Detailed Abalations and Analysis

In this section, we perform ablations and provide a detailed analysis of Imagen.

## D.1 Pre-trained Text Encoders

We explore several families of pre-trained text encoders: BERT [15], T5 [52], and CLIP [49]. There are several key differences between these encoders. BERT is trained on a smaller text-only corpus (approximately 20 GB, Wikipedia and BooksCorpus [84]) with a masking objective, and has relatively small model variants (upto 340M parameters). T5 is trained on a much larger C4 text-only corpus (approximately 800 GB) with a denoising objective, and has larger model variants (up to 11B parameters). The CLIP model[5] is trained on an image-text corpus with an image-text contrastive objective. For T5 we use the encoder part for the contextual embeddings. For CLIP, we use the penultimate layer of the text encoder to get contextual embeddings. Note that we freeze the weights of these text encoders (i.e., we use off the shelf text encoders, without any fine-tuning on the text-to-image generation task). We explore a variety of model sizes for these text encoders.

We train a $64 \times 64$, 300M parameter diffusion model, conditioned on the text embeddings generated from BERT (base, and large), T5 (small, base, large, XL, and XXL), and CLIP (ViT-L/14). We observe that scaling the size of the language model text encoders generally results in better image-text

[5] https://github.com/openai/CLIP/blob/main/model-card.md

| Category | Description | Examples |
|---|---|---|
| Colors | Ability to generate objects with specified colors. | "A blue colored dog." "A black apple and a green backpack." |
| Counting | Ability to generate specified number of objects. | "Three cats and one dog sitting on the grass." "Five cars on the street." |
| Conflicting | Ability to generate conflicting interactions b/w objects. | "A horse riding an astronaut." "A panda making latte art." |
| DALL-E [53] | Subset of challenging prompts from [53]. | "A triangular purple flower pot." "A cross-section view of a brain." |
| Description | Ability to understand complex and long text prompts describing objects. | "A small vessel propelled on water by oars, sails, or an engine." "A mechanical or electrical device for measuring time." |
| Marcus et al. [38] | Set of challenging prompts from [38]. | "A pear cut into seven pieces arranged in a ring." "Paying for a quarter-sized pizza with a pizza-sized quarter." |
| Misspellings | Ability to understand misspelled prompts. | "Rbefraigerator." "Tcennis rpacket." |
| Positional | Ability to generate objects with specified spatial positioning. | "A car on the left of a bus." "A stop sign on the right of a refrigerator." |
| Rare Words | Ability to understand rare words[3]. | "Artophagous." "Octothorpe." |
| Reddit | Set of challenging prompts from DALLE-2 Reddit[4]. | "A yellow and black bus cruising through the rainforest." "A medieval painting of the wifi not working." |
| Text | Ability to generate quoted text. | "A storefront with 'Deep Learning' written on it." "A sign that says 'Text to Image'." |

Table A.1: Description and examples of the 11 categories in DrawBench.

alignment as captured by the CLIP score as a function of number of training steps (see Fig. A.6). One can see that the best CLIP scores are obtained with the T5-XXL text encoder.

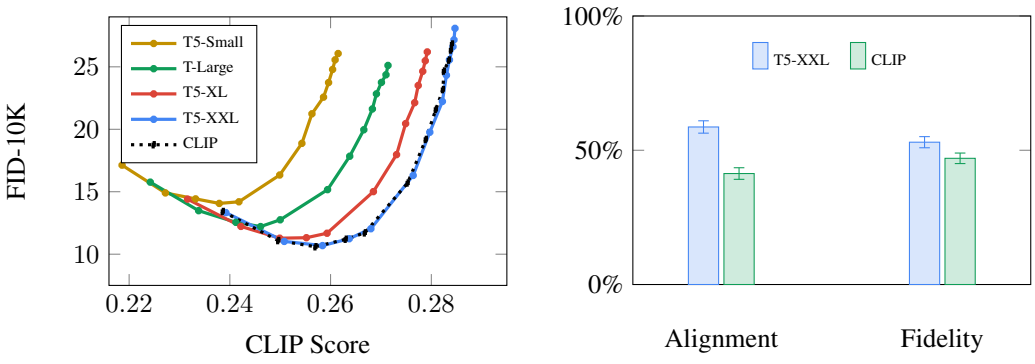
Since guidance weights are used to control image quality and text alignment, we also report ablation results using curves that show the trade-off between CLIP and FID scores as a function of the guidance weights (see Fig. A.5a). We observe that larger variants of T5 encoder results in both better image-text alignment, and image fidelity. This emphasizes the effectiveness of large frozen text encoders for text-to-image models. Interestingly, we also observe that the T5-XXL encoder is on-par with the CLIP encoder when measured with CLIP and FID-10K on MS-COCO.

**T5-XXL vs CLIP on DrawBench**: We further compare T5-XXL and CLIP on DrawBench to perform a more comprehensive comparison of the abilities of these two text encoders. In our initial evaluations we observed that the 300M parameter models significantly underperformed on DrawBench. We believe this is primarily because DrawBench prompts are considerably more difficult than MS-COCO prompts.

In order to perform a meaningful comparison, we train $64\times64$ 1B parameter diffusion models with T5-XXL and CLIP text encoders for this evaluation. Fig. A.5b shows the results. We find that raters are considerably more likely to prefer the generations from the model trained with the T5-XXL encoder over the CLIP text encoder, especially for image-text alignment. This indicates that language models are better than text encoders trained on image-text contrastive objectives in encoding complex and compositional text prompts. Fig. A.7 shows the category specific comparison between the two models. We observe that human raters prefer T5-XXL samples over CLIP samples in all 11 categories for image-text alignment demonstrating the effectiveness of large language models as text encoders for text to image generation.

### D.2 Classifier-free Guidance and the Alignment-Fidelity Trade-off

We observe that classifier-free guidance [27] is a key contributor to generating samples with strong image-text alignment, this is also consistent with the observations of [53, 54]. There is typically a trade-off between image fidelity and image-text alignment, as we iterate over the guidance weight. While previous work has typically used relatively small guidance weights, Imagen uses relatively large guidance weights for all three diffusion models. We found this to yield a good balance of sample quality and alignment. However, naive use of large guidance weights often produces relatively poor




(a) Pareto curves comparing various text encoders.

(b) Comparing T5-XXL and CLIP on DrawBench.

Figure A.5: Comparison between text encoders for text-to-image generation. For Fig. A.5a, we sweep over guidance values of $[1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

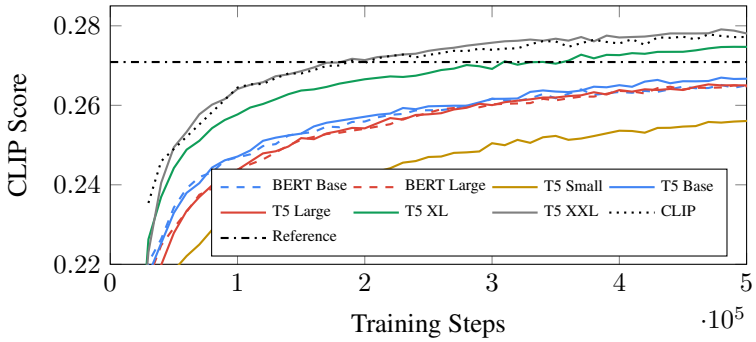


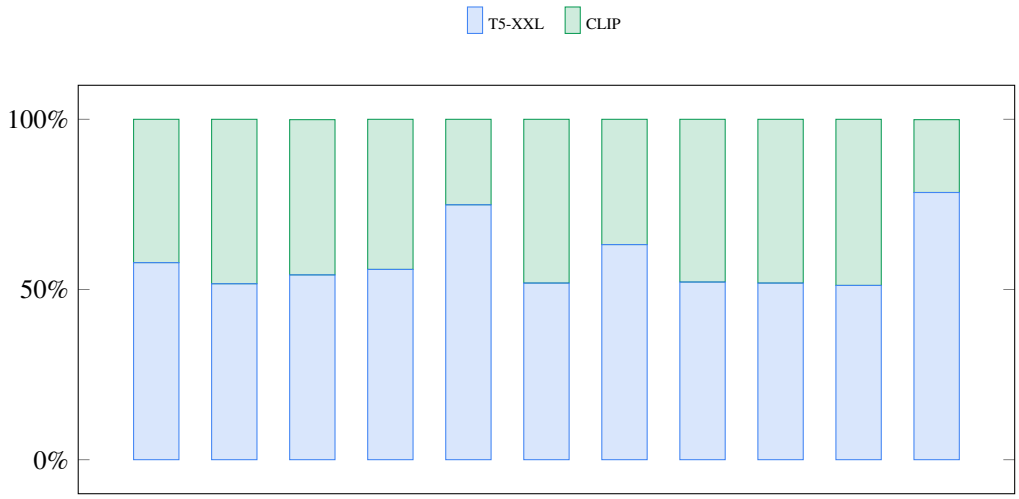

Figure A.6: Training convergence comparison between text encoders for text-to-image generation.

results. To enable the effective use of larger guidance we introduce several innovations, as described below.

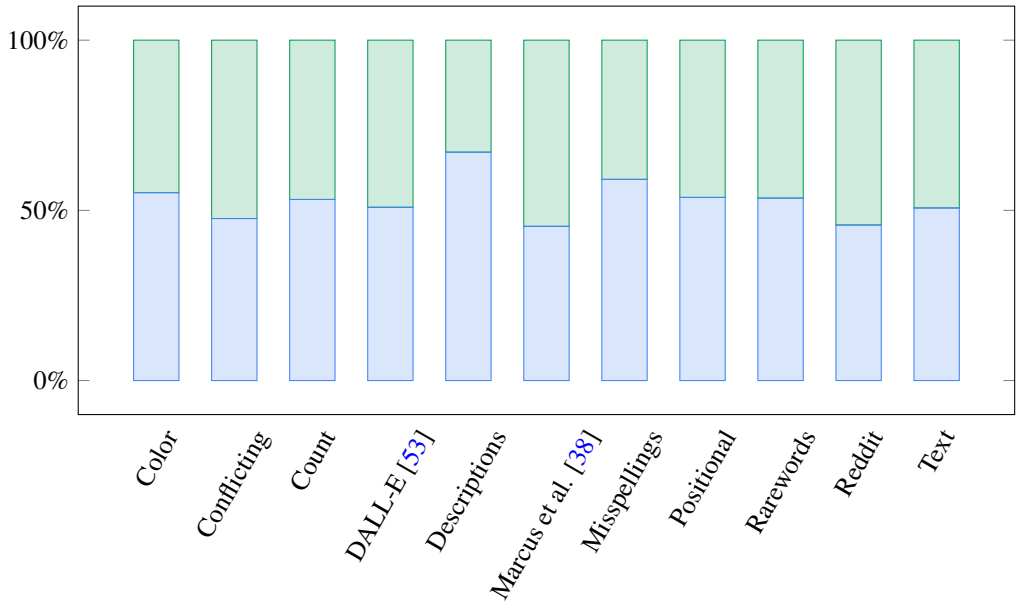
**Thresholding Techniques**: First, we compare various thresholding methods used with classifier-free guidance. Fig. A.8 compares the CLIP vs. FID-10K score pareto frontiers for various thresholding methods of the base text-to-image $64 \times 64$ model. We observe that our dynamic thresholding technique results in significantly better CLIP scores, and comparable or better FID scores than the static thresholding technique for a wide range of guidance weights. Fig. A.9 shows qualitative samples for thresholding techniques.

**Guidance for Super-Resolution**: We further analyze the impact of classifier-free guidance for our $64 \times 64 \rightarrow 256 \times 256$ model. Fig. A.11a shows the pareto frontiers for CLIP vs. FID-10K score for the $64 \times 64 \rightarrow 256 \times 256$ super-resolution model. $\text{aug\_level}$ specifies the level of noise augmentation applied to the input low-resolution image during inference ($\text{aug\_level} = 0$ means no noise). We observe that $\text{aug\_level} = 0$ gives the best FID score for all values of guidance weight. Furthermore, for all values of $\text{aug\_level}$, we observe that FID improves considerably with increasing guidance weight upto around $7 - 10$. While generation using larger values of $\text{aug\_level}$ gives slightly worse FID, it allows more varied range of CLIP scores, suggesting more diverse generations by the super-resolution model. In practice, for our best samples, we generally use $\text{aug\_level}$ in $[0.1, 0.3]$. Using large values of $\text{aug\_level}$ and high guidance weights for the super-resolution models, Imagen can create different variations of a given $64 \times 64$ image by altering the prompts to the super-resolution models (See Fig. A.12 for examples).

**Impact of Conditioning Augmentation**: Fig. A.11b shows the impact of training super-resolution models with noise conditioning augmentation. Training with no noise augmentation generally results in worse CLIP and FID scores, suggesting noise conditioning augmentation is critical to attaining best sample quality similar to prior work [29]. Interestingly, the model trained without noise augmentation has much less variations in CLIP and FID scores across different guidance weights compared to




(a) Alignment




(b) Fidelity

Figure A.7: T5-XXL vs. CLIP text encoder on DrawBench a) image-text alignment, and b) image fidelity.

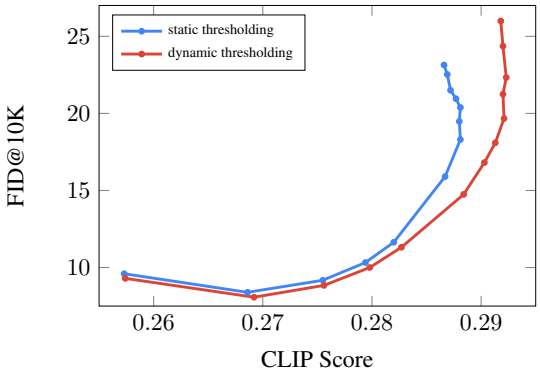



Figure A.8: CLIP Score vs FID trade-off across various $\hat{\mathbf{x}}_0$ thresholding methods for the 64×64 model. We sweep over guidance values of $[1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 6, 7, 8, 9, 10]$.

(a) No thresholding. (b) Static thresholding. (c) Dynamic thresholding.

Figure A.9: Thresholding techniques on $256 \times 256$ samples for "A photo of an astronaut riding a horse." Guidance weights increase from 1 to 5 as we go from top to bottom. No thresholding results in poor images with high guidance weights. Static thresholding is an improvement but still leads to oversaturated samples. Our dynamic thresholding leads to the highest quality images. See Fig. A.10 for more qualitative comparison.

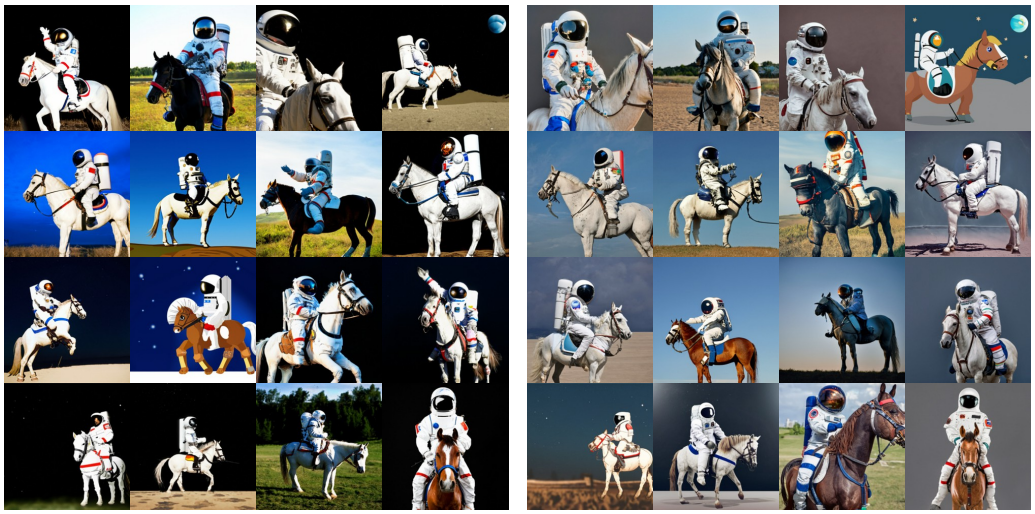
the model trained with conditioning augmentation. We hypothesize that this is primarily because strong noise augmented training reduces the low-resolution image conditioning signal considerably, encouraging higher degree of dependence on conditioned text for the model.

### D.3 Impact of Model Size

Fig. A.13b plots the CLIP-FID score trade-off curves for various model sizes of the $64 \times 64$ text-to-image U-Net model. We train each of the models with a batch size of 2048, and 400K training steps. As we scale from 300M parameters to 2B parameters for the U-Net model, we obtain better trade-off curves with increasing model capacity. Interestingly, scaling the frozen text encoder model size yields more improvement in model quality over scaling the U-Net model size. Scaling with a frozen text encoder is also easier since the text embeddings can be computed and stored offline during training.

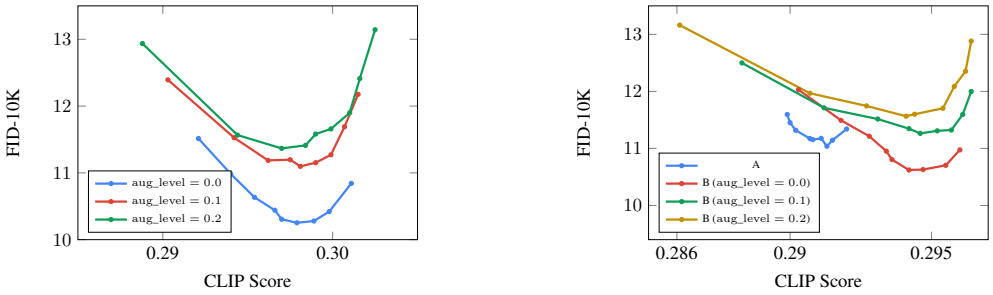
#### D.3.1 Impact of Text Conditioning Schemas

We ablate various schemas for conditioning the frozen text embeddings in the base $64 \times 64$ text-to-image diffusion model. Fig. A.13a compares the CLIP-FID pareto curves for mean pooling, attention pooling, and cross attention. We find using any pooled embedding configuration (mean or attention pooling) performs noticeably worse compared to attending over the sequence of contextual embeddings in the attention layers. We implement the cross attention by concatenating the text



(a) Samples using static thresholding. (b) Samples using dynamic thresholding ($p = 99.5$)

Figure A.10: Static vs. dynamic thresholding on non-cherry picked $256 \times 256$ samples using a guidance weight of 5 for both the base model and the super-resolution model, using the same random seed. The text prompt used for these samples is "A photo of an astronaut riding a horse." When using high guidance weights, static thresholding often leads to oversaturated samples, while our dynamic thresholding yields more natural looking images.




(a) Comparison between different values of aug_level. (b) Comparison between training with no noise augmentation "A" vs noise augmentation "B"

Figure A.11: CLIP vs FID-10K pareto curves showing the impact of noise augmentation on our $64 \times 64 \rightarrow 256 \times 256$ model. For each study, we sweep over guidance values of $[1, 3, 5, 7, 8, 10, 12, 15, 18]$

embedding sequence to the key-value pairs of each self-attention layer in the base $64 \times 64$ and $64 \times 64 \rightarrow 256 \times 256$ models. For our $256 \times 256 \rightarrow 1024 \times 1024$ model, since we have no self-attention layers, we simply added explicit cross-attention layers to attend over the text embeddings. We found this to improve both fidelity and image-text alignment with minimal computational costs.

### D.3.2 Comparison of U-Net vs Efficient U-Net

We compare the performance of U-Net with our new Efficient U-Net on the task of $64 \times 64 \rightarrow 256 \times 256$ super-resolution task. Fig. A.14 compares the training convergence of the two architectures. We observe that Efficient U-Net converges significantly faster than U-Net, and obtains better performance overall. Our Efficient U-Net is also $\times 2 - 3$ faster at sampling.

## E Comparison to GLIDE and DALL-E 2

Fig. A.15 shows category wise comparison between Imagen and DALL-E 2 [54] on DrawBench. We observe that human raters clearly prefer Imagen over DALL-E 2 in 7 out of 11 categories for text alignment. For sample fidelity, they prefer Imagen over DALL-E 2 in all 11 categories. Figures A.17

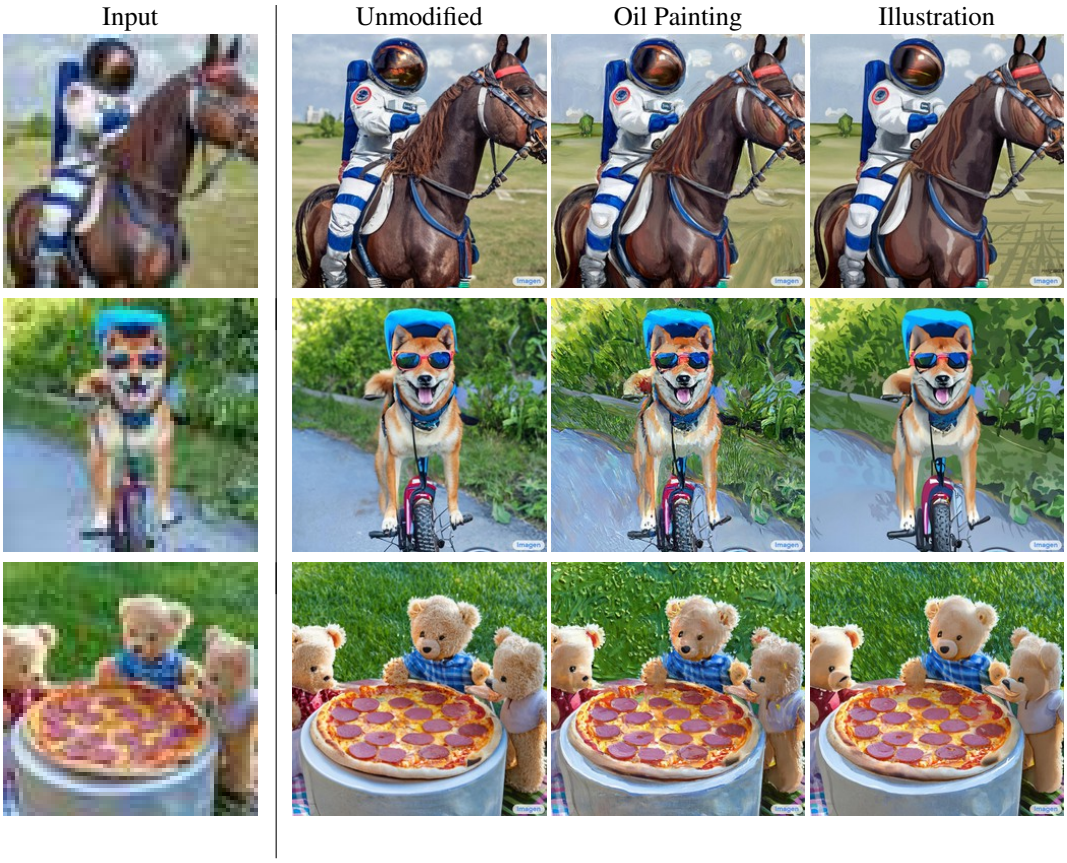
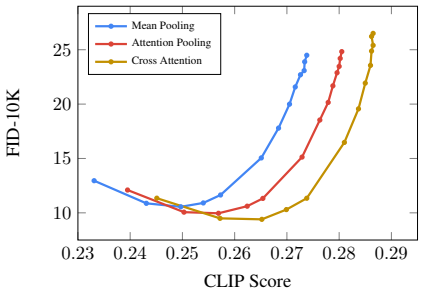
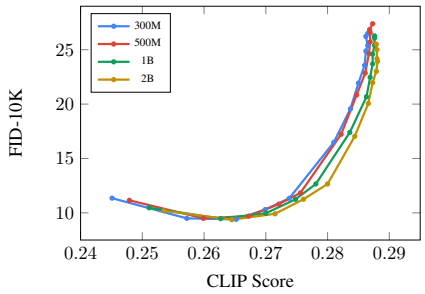



Figure A.12: Super-resolution variations for some $64 \times 64$ generated images. We first generate the $64 \times 64$ image using "A photo of ... .". Given generated $64 \times 64$ images, we condition both the super-resolution models on different prompts in order to generate different upsampled variations. e.g. for oil painting we condition the super-resolution models on the prompt "An oil painting of ... .". Through a combination of large guidance weights and $\text{aug\_level} = 0.3$ for both super-res models we can generate different styles based on the style query through text.

to A.21 show few qualitative comparisons between Imagen and DALL-E 2 samples used for this human evaluation study. Some of the categories where Imagen has a considerably larger preference over DALL-E 2 include Colors, Positional, Text, DALL-E and Descriptions. The authors in [54] identify some of these limitations of DALL-E 2, specifically they observe that DALLE-E 2 is worse than GLIDE [41] in binding attributes to objects such as colors, and producing coherent text from the input prompt (cf. the discussion of limitations in [54]). To this end, we also perform quantitative and qualitative comparison with GLIDE [41] on DrawBench. See Fig. A.16 for category wise human evaluation comparison between Imagen and GLIDE. See Figures A.22 to A.26 for qualitative comparisons. Imagen outperforms GLIDE on 8 out of 11 categories on image-text alignment, and 10 out of 11 categories on image fidelity. We observe that GLIDE is considerably better than DALL-E 2 in binding attributes to objects corroborating the observation by [54].




(a) Comparison between different text encoders.




(b) Comparison between different model sizes.

Figure A.13: CLIP vs FID-10K pareto curves for different ablation studies for the base $64 \times 64$ model. For each study, we sweep over guidance values of $[1, 1.25, 1.5, 1.75, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

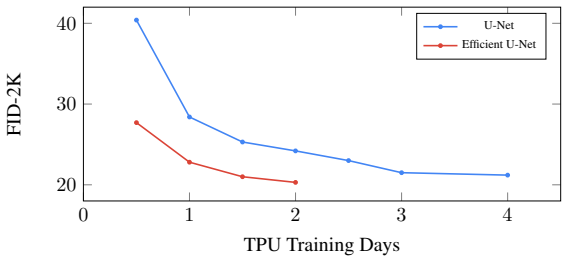
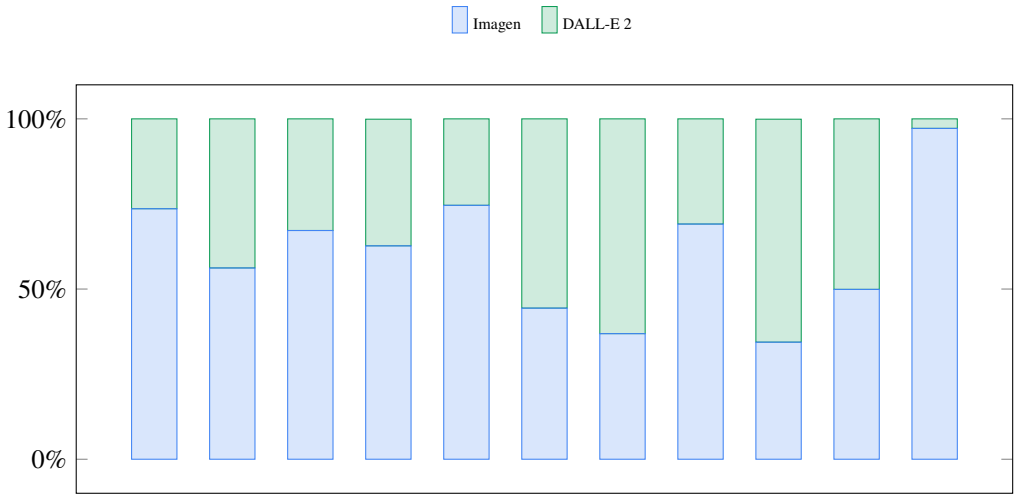
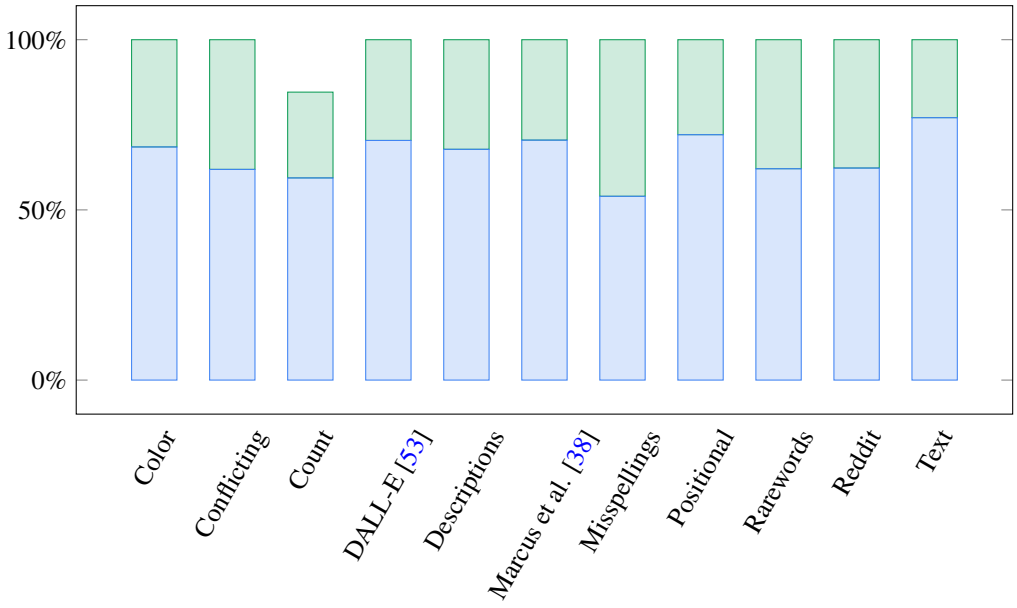



Figure A.14: Comparison of convergence speed of U-Net vs Efficient U-Net on the $64 \times 64 \rightarrow 256 \times 256$ super-resolution task.

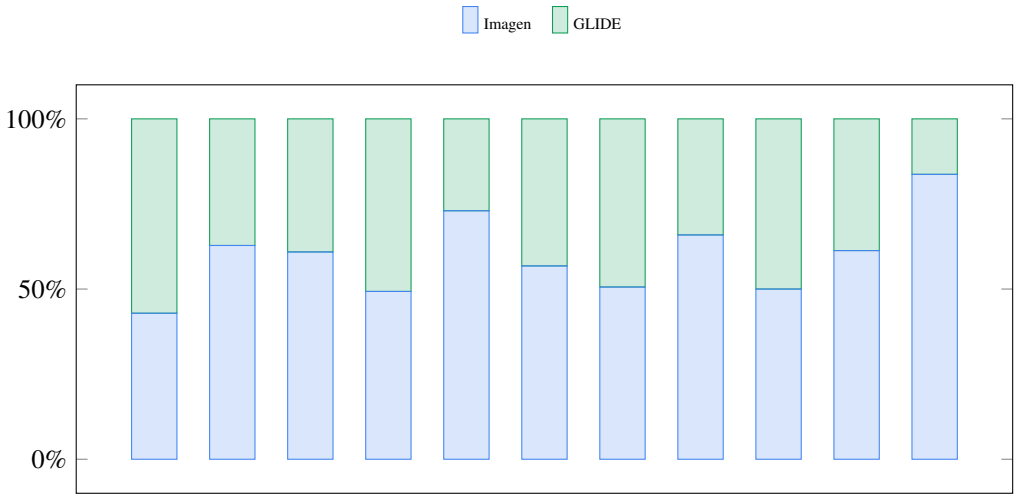



(a) Alignment

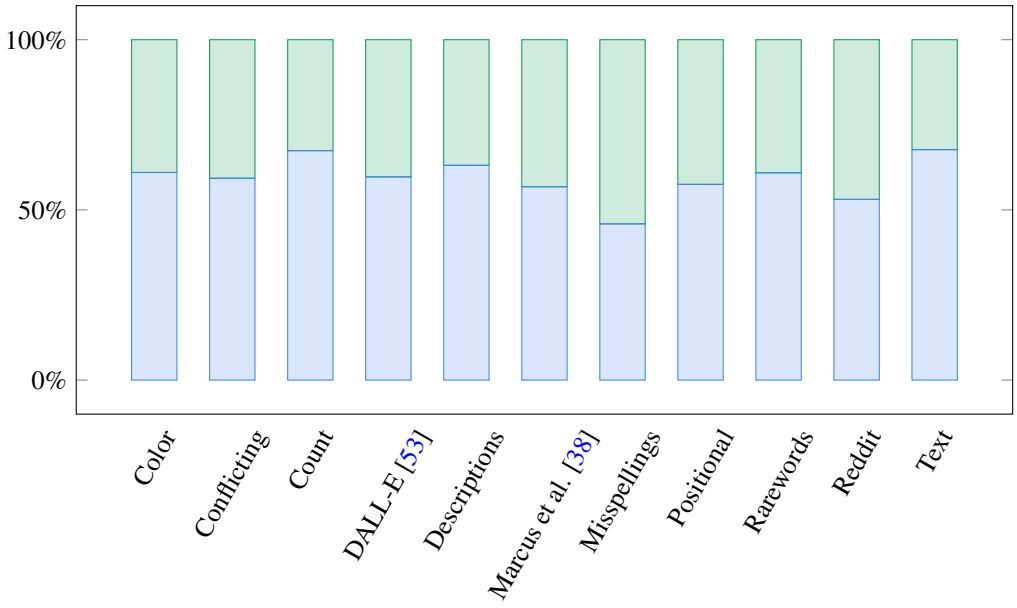



(b) Fidelity

Figure A.15: Imagen vs DALL-E 2 on DrawBench a) image-text alignment, and b) image fidelity.




(a) Alignment




(b) Fidelity

Figure A.16: Imagen vs GLIDE on DrawBench a) image-text alignment, and b) image fidelity.

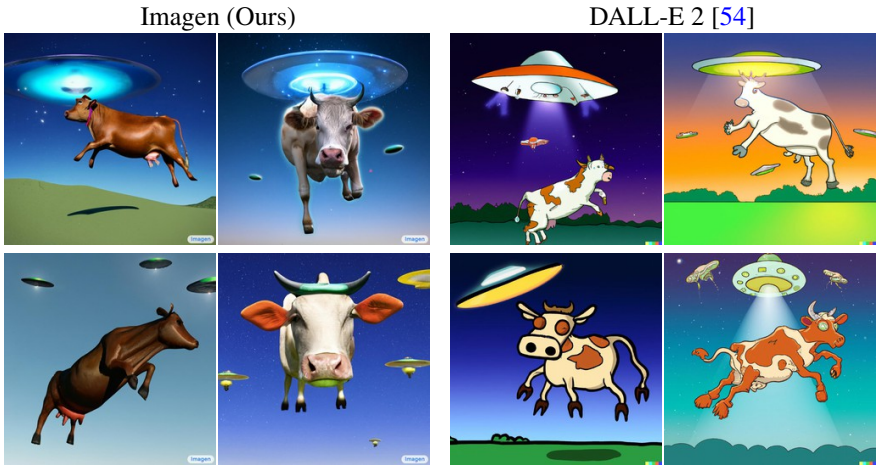


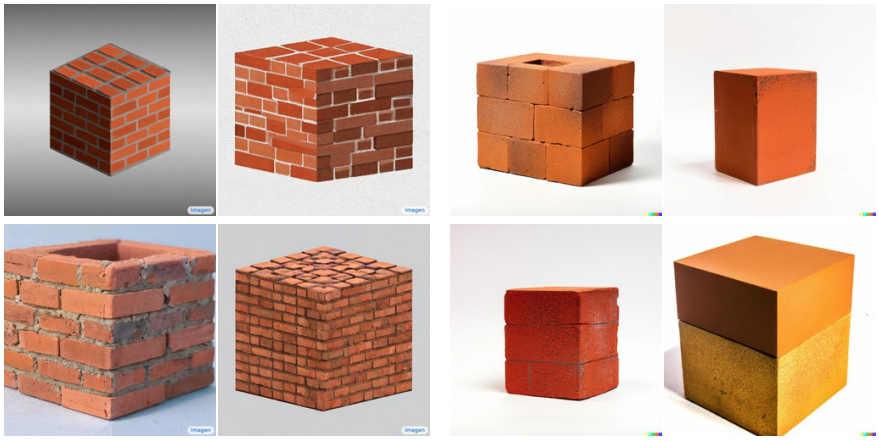

Figure A.17: Example qualitative comparisons between Imagen and DALL-E 2 [54] on DrawBench prompts from Reddit category.


Figure A.18: Example qualitative comparisons between Imagen and DALL-E 2 [54] on DrawBench prompts from Colors category. We observe that DALL-E 2 generally struggles with correctly assigning the colors to the objects especially for prompts with more than one object.


Figure A.19: Example qualitative comparisons between Imagen and DALL-E 2 [54] on DrawBench prompts from Conflicting category. We observe that both DALL-E 2 and Imagen struggle generating well aligned images for this category. However, Imagen often generates some well aligned samples, e.g. “A panda making latte art.”.

Imagen (Ours) DALL-E 2 [54]

A couple of glasses are sitting on a table.



A cube made of brick. A cube with the texture of brick.

Figure A.20: Example qualitative comparisons between Imagen and DALL-E 2 [54] on DrawBench prompts from DALL-E category.


New York Skyline with Hello World written with fireworks on the sky.


A storefront with Text to Image written on it.

Figure A.21: Example qualitative comparisons between Imagen and DALL-E 2 [54] on DrawBench prompts from Text category. Imagen is significantly better than DALL-E 2 in prompts with quoted text.

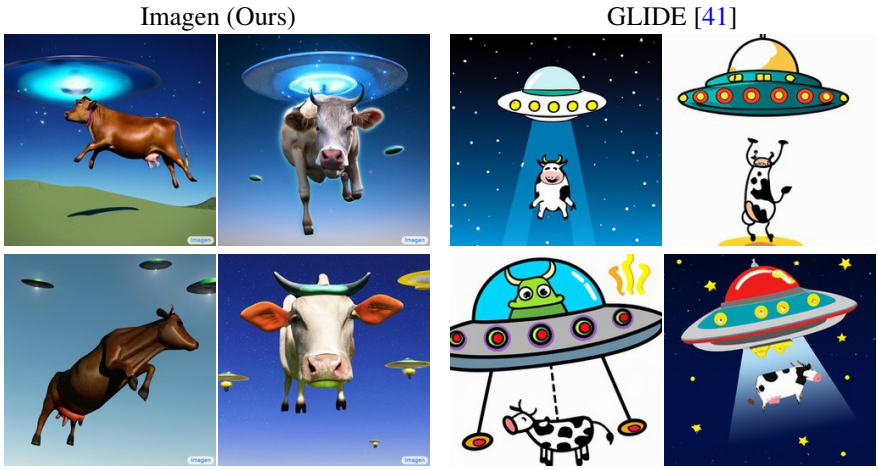



Figure A.22: Example qualitative comparisons between Imagen and GLIDE [41] on DrawBench prompts from Reddit category.


Figure A.23: Example qualitative comparisons between Imagen and GLIDE [41] on DrawBench prompts from Colors category. We observe that GLIDE is better than DALL-E 2 in assigning the colors to the objects.

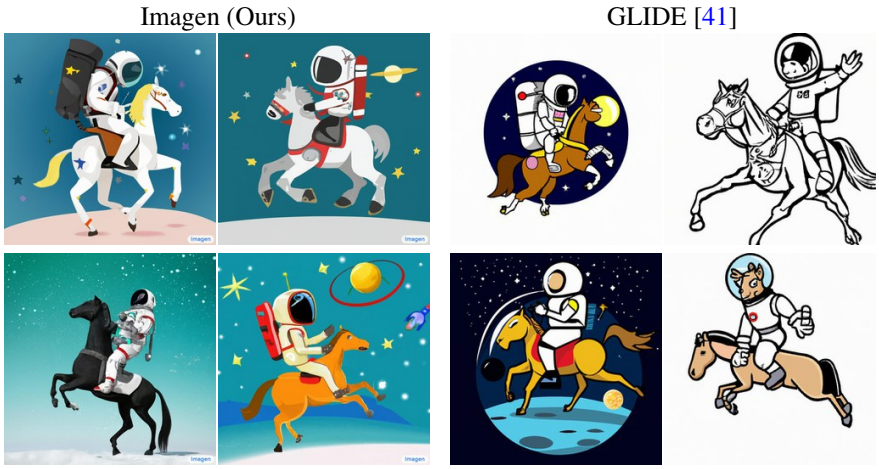



Figure A.24: Example qualitative comparisons between Imagen and GLIDE [41] on DrawBench prompts from Conflicting category.


Figure A.25: Example qualitative comparisons between Imagen and GLIDE [41] on DrawBench prompts from DALL-E category.


Figure A.26: Example qualitative comparisons between Imagen and GLIDE [41] on DrawBench prompts from Text category. Imagen is significantly better than GLIDE too in prompts with quoted text.

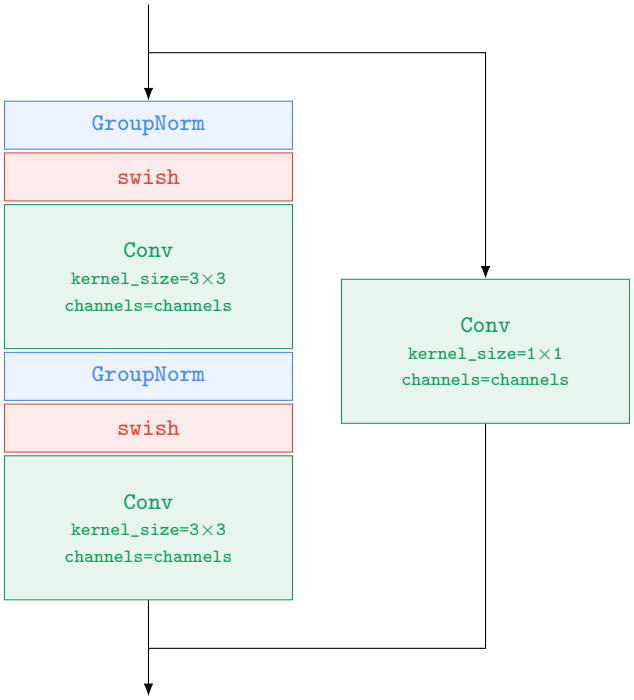



Figure A.27: Efficient U-Net `ResNetBlock`. The `ResNetBlock` is used both by the `DBlock` and `UBlock`. Hyperparameter of the `ResNetBlock` is the number of channels `channels: int`.

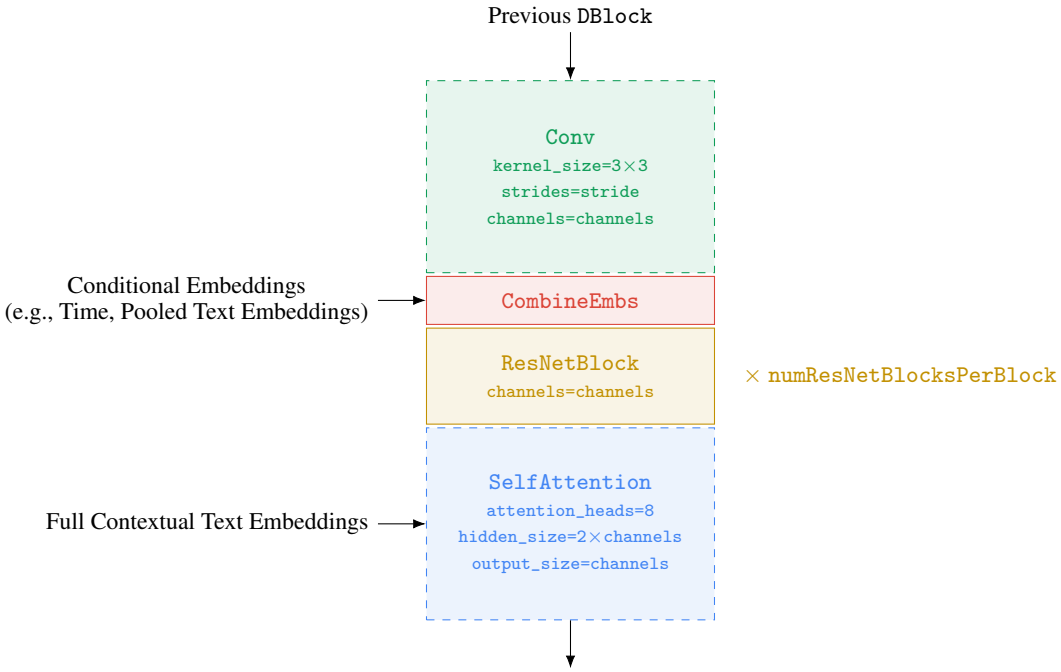



Figure A.28: Efficient UNet `DBlock`. Hyperparameters of `DBlock` are: the stride of the block if there is downsampling `stride: Optional[Tuple[int, int]]`, number of `ResNetBlock` per `DBlock` `numResNetBlocksPerBlock: int`, and number of channels `channels: int`. The dashed lined blocks are optional, e.g., not every `DBlock` needs to downsample or needs self-attention.

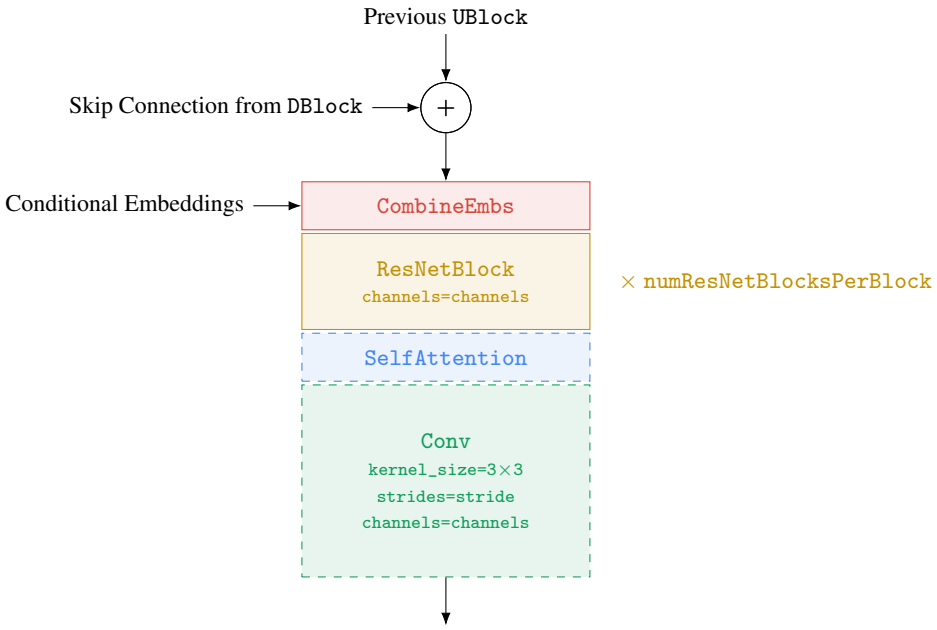



Figure A.29: Efficient U-Net `UBlock`. Hyperparameters of `UBlock` are: the stride of the block if there is upsampling `stride: Optional[Tuple[int, int]]`, number of `ResNetBlock` per `DBlock` `numResNetBlocksPerBlock: int`, and number of channels `channels: int`. The dashed lined blocks are optional, e.g., not every `UBlock` needs to upsample or needs self-attention.

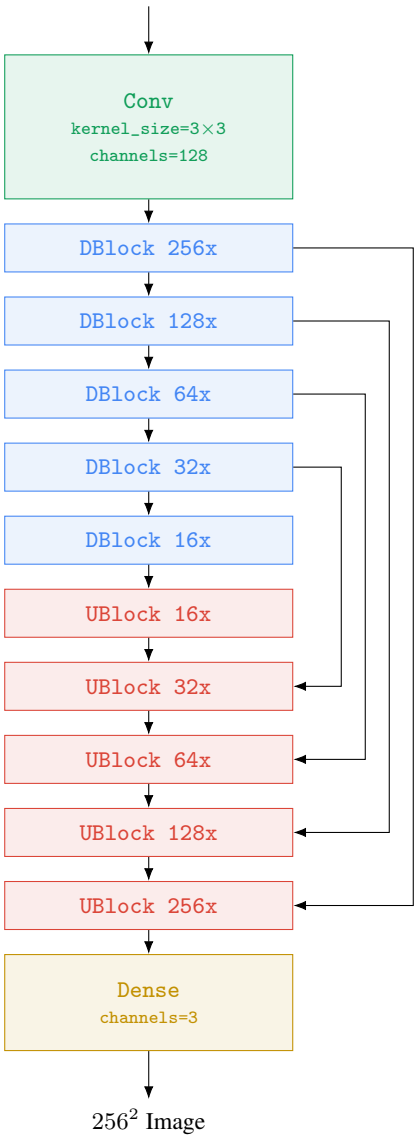



Figure A.30: Efficient U-Net architecture for $64^2 \rightarrow 256^2$.

```
def sample():
  for t in reversed(range(T)):
    # Forward pass to get x0_t from z_t.
    x0_t = nn(z_t, t)

    # Static thresholding.
    x0_t = jnp.clip(x0_t, -1.0, 1.0)




    # Sampler step.
    z_tm1 = sampler_step(x0_t, z_t, t)
    z_t = z_tm1
  return x0_t
```

(a) Implementation for static thresholding.

```
def sample(p: float):
  for t in reversed(range(T)):
    # Forward pass to get x0_t from z_t.
    x0_t = nn(z_t, t)

    # Dynamic thresholding (ours).
    s = jnp.percentile(
        jnp.abs(x0_t), p,
        axis=tuple(range(1, x0_t.ndim)))
    s = jnp.max(s, 1.0)
    x0_t = jnp.clip(x0_t, -s, s) / s

    # Sampler step.
    z_tm1 = sampler_step(x0_t, z_t, t)
    z_t = z_tm1
  return x0_t
```

(b) Implementation for dynamic thresholding.

Figure A.31: Pseudo code implementation comparing static thresholding and dynamic thresholding.

```
def train_step(
    x_lr: jnp.ndarray, x_hr: jnp.ndarray):
  # Add augmentation to the low-resolution image.
  aug_level = jnp.random.uniform(0.0, 1.0)
  x_lr = apply_aug(x_lr, aug_level)

  # Diffusion forward process.
  t = jnp.random.uniform(0.0, 1.0)
  z_t = forward_process(x_hr, t)

  Optimize loss(x_hr, nn(z_t, x_lr, t, aug_level))
```

(a) Training using conditioning augmentation.

```
def sample(aug_level: float, x_lr: jnp.ndarray):
  # Add augmentation to the low-resolution image.
  x_lr = apply_aug(x_lr, aug_level)

  for t in reversed(range(T)):
    x_hr_t = nn(z_t, x_lr, t, aug_level)

    # Sampler step.
    z_tm1 = sampler_step(x_hr_t, z_t, t)
    z_t = z_tm1
  return x_hr_t
```

(b) Sampling using conditioning augmentation.

Figure A.32: Pseudo-code implementation for training and sampling using conditioning augmentation. Text conditioning has not been shown for brevity.

# F Implementation Details

## F.1 $64 \times 64$

**Architecture**: We adapt the architecture used in [16]. We use larger *embed_dim* for scaling up the architecture size. For conditioning on text, we use text cross attention at resolutions $[32, 16, 8]$ as well as attention pooled text embedding.

**Optimizer**: We use the Adafactor optimizer for training the base model. We use the default optax.adafactor parameters. We use a learning rate of 1e-4 with 10000 linear warmup steps.

**Diffusion**: We use the cosine noise schedule similar to [40]. We train using continuous time steps $t \sim \mathcal{U}(0, 1)$.

```
# 64 X 64 model.
architecture = {
     "attn_resolutions": [32, 16, 8],
     "channel_mult": [1, 2, 3, 4],
     "dropout": 0,
     "embed_dim": 512,
     "num_res_blocks": 3,
     "per_head_channels": 64,
     "res_block_type": "biggan",
     "text_cross_attn_res": [32, 16, 8],
     "feature_pooling_type": "attention",
     "use_scale_shift_norm": True,
 }

 learning_rate = optax.warmup_cosine_decay_schedule(
     init_value=0.0,
     peak_value=1e-4,
     warmup_steps=10000,
     decay_steps=2500000,
     end_value=2500000)

 optimizer = optax.adafactor(lrs=learning_rate, weight_decay=0)
```

```
diffusion_params = {
  "continuous_time": True,
  "schedule": {
    "name": "cosine",
  }
}
```

### F.2 $64 \times 64 \rightarrow 256 \times 256$

**Architecture**: Below is the architecture specification for our $64 \times 64 \rightarrow 256 \times 256$ super-resolution model. We use an Efficient U-Net architecture for this model.

**Optimizer**: We use the standard Adam optimizer with 1e-4 learning rate, and 10000 warmup steps.

**Diffusion**: We use the same cosine noise schedule as the base $64 \times 64$ model. We train using continuous time steps $t \sim \mathcal{U}(0, 1)$.

```
architecture = {
    "dropout": 0.0,
    "feature_pooling_type": "attention",
    "use_scale_shift_norm": True,
    "blocks": [
        {
          "channels": 128,
          "strides": (2, 2),
          "kernel_size": (3, 3),
          "num_res_blocks": 2,
        },
        {
          "channels": 256,
          "strides": (2, 2),
          "kernel_size": (3, 3),
          "num_res_blocks": 4,
        },
        {
          "channels": 512,
          "strides": (2, 2),
          "kernel_size": (3, 3),
          "num_res_blocks": 8,
        },
        {
          "channels": 1024,
          "strides": (2, 2),
          "kernel_size": (3, 3),
          "num_res_blocks": 8,
          "self_attention": True,
          "text_cross_attention": True,
          "num_attention_heads": 8
        }
    ]
}

learning_rate = optax.warmup_cosine_decay_schedule(
    init_value=0.0,
    peak_value=1e-4,
    warmup_steps=10000,
    decay_steps=2500000,
    end_value=2500000)

optimizer = optax.adam(
    lrs=learning_rate, b1=0.9, b2=0.999, eps=1e-8, weight_decay=0)

diffusion_params = {
  "continuous_time": True,
  "schedule": {
    "name": "cosine",
  }
}
```

### F.3 $256 \times 256 \rightarrow 1024 \times 1024$

**Architecture**: Below is the architecture specification for our $256 \times 256 \rightarrow 1024 \times 1024$ super-resolution model. We use the same configuration as the $64 \times 64 \rightarrow 256 \times 256$ super-resolution model, except we do not use self-attention layers but rather have cross-attention layers (to the text embeddings).

**Optimizer**: We use the standard Adam optimizer with 1e-4 learning rate, and 10000 linear warmup steps.

**Diffusion**: We use the 1000 step linear noise schedule with start and end set to 1e-4 and 0.02 respectively. We train using continuous time steps $t \sim \mathcal{U}(0, 1)$.

```
"dropout": 0.0,
"feature_pooling_type": "attention",
"use_scale_shift_norm": true,
"blocks"=[
    {
      "channels": 128,
      "strides": (2, 2),
      "kernel_size": (3, 3),
      "num_res_blocks": 2,
    },
    {
      "channels": 256,
      "strides": (2, 2),
      "kernel_size": (3, 3),
      "num_res_blocks": 4,
    },
    {
      "channels": 512,
      "strides": (2, 2),
      "kernel_size": (3, 3),
      "num_res_blocks": 8,
    },
    {
      "channels": 1024,
      "strides": (2, 2),
      "kernel_size": (3, 3),
      "num_res_blocks": 8,
      "text_cross_attention": True,
      "num_attention_heads": 8
    }
]
```