StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis

Axel Sauer [1 2] **Tero Karras** [2] **Samuli Laine** [2] **Andreas Geiger** [1] **Timo Aila** [2]

Abstract

Text-to-image synthesis has recently seen significant progress thanks to large pretrained language models, large-scale training data, and the introduction of scalable model families such as diffusion and autoregressive models. However, the best-performing models require iterative evaluation to generate a single sample. In contrast, generative adversarial networks (GANs) only need a single forward pass. They are thus much faster, but they currently remain far behind the state-of-the-art in large-scale text-to-image synthesis. This paper aims to identify the necessary steps to regain competitiveness. Our proposed model, StyleGAN-T, addresses the specific requirements of large-scale text-to-image synthesis, such as large capacity, stable training on diverse datasets, strong text alignment, and controllable variation vs. text alignment tradeoff. StyleGAN-T significantly improves over previous GANs and outperforms distilled diffusion models — the previous state-of-the-art in fast text-to-image synthesis — in terms of sample quality and speed.

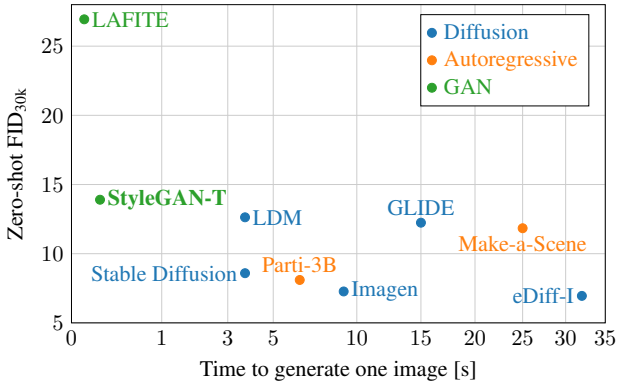

Figure 1. **Quality vs. speed** in large-scale text-to-image synthesis. StyleGAN-T greatly narrows the quality gap between GANs and other model families while generating samples at a rate of 10 FPS on an NVIDIA A100. The y-axis corresponds to zero-shot FID on MS COCO at 256×256 resolution; lower is better.

1. Introduction

In text-to-image synthesis, novel images are generated based on text prompts. The state-of-the-art in this task has recently taken dramatic leaps forward thanks to two key ideas. First, using a large pretrained language model as an encoder for the prompts makes it possible to condition the synthesis based on general language understanding (Ramesh et al., 2022; Saharia et al., 2022). Second, using large-scale training data consisting of hundreds of millions of image-caption pairs (Schuhmann et al., 2022) allows the models to synthesize almost anything imaginable.

Training datasets continue to increase rapidly in size and coverage. Consequently, text-to-image models must be scalable to a large capacity to absorb the training data. Recent successes in large-scale text-to-image generation have been driven by diffusion models (DM) (Ramesh et al., 2022; Saharia et al., 2022; Rombach et al., 2022) and autoregressive models (ARM) (Zhang et al., 2021; Yu et al., 2022; Gafni et al., 2022) that seem to have this property built in, along with the ability to deal with highly multi-modal data.

Interestingly, generative adversarial networks (GAN) (Goodfellow et al., 2014) — the dominant family of generative models in smaller and less diverse datasets — have not been particularly successful in this task (Zhou et al., 2022). Our goal is to show that they can regain competitiveness.

The primary benefits offered by GANs are inference speed and control of the synthesized result via latent space manipulations. StyleGAN (Karras et al., 2019; 2020; 2021) in particular has a thoroughly studied latent space, which allows principled control of generated images (Bermano et al., 2022; Härkönen et al., 2020; Shen et al., 2020; Abdal et al., 2021; Kafri et al., 2022). While there has been notable progress in speeding up DMs (Salimans & Ho, 2022; Karras et al., 2022; Lu et al., 2022), they are still far behind GANs that require only a single forward pass.

[1]University of Tübingen, Tübingen AI Center [2]NVIDIA. Correspondence to: Axel Sauer <a.sauer@uni-tuebingen.de>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

We draw motivation from the observation that GANs lagged similarly behind diffusion models in ImageNet (Deng et al., 2009; Dhariwal & Nichol, 2021) synthesis until the discriminator architecture was redesigned in StyleGAN-XL (Sauer et al., 2021; 2022), which allowed GANs to close the gap. In Section 3, we start from StyleGAN-XL and revisit the generator and discriminator architectures, considering the requirements specific to the large-scale text-to-image task: large capacity, extremely diverse datasets, strong text alignment, and controllable variation vs. text alignment tradeoff.

We have a fixed training budget of 4 weeks on 64 NVIDIA A100s available for training our final model at scale. This constraint forces us to set priorities because the budget is likely insufficient for state-of-the-art, high-resolution results (CompVis, 2022). While the ability of GANs to scale to high resolutions is well known (Wang et al., 2018; Karras et al., 2020), successful scaling to the large-scale text-to-image task remains undocumented. We thus focus primarily on solving this task in lower resolutions, dedicating only a limited budget to the super-resolution stages.

Our StyleGAN-T achieves a better zero-shot MS COCO FID (Lin et al., 2014; Heusel et al., 2017) than current state-of-the-art diffusion models at a resolution of 64×64. At 256×256, StyleGAN-T halves the zero-shot FID previously achieved by a GAN but continues to trail SOTA diffusion models. The key benefits of StyleGAN-T include its fast inference speed and smooth latent space interpolation in the context of text-to-image synthesis, illustrated in Fig. 1 and Fig. 2, respectively. Our implementation is available at https://github.com/autonomousvision/stylegan-t.

2. StyleGAN-XL

Our architecture design is based on StyleGAN-XL (Sauer et al., 2022) that — similar to the original StyleGAN (Karras et al., 2019) — first processes the normally distributed input latent code $\mathbf{z}$ by a mapping network to produce an intermediate latent code $\mathbf{w}$. This intermediate latent is then used to modulate the convolution layers in a synthesis network using the weight demodulation technique introduced in StyleGAN2 (Karras et al., 2020). The synthesis network of StyleGAN-XL uses the alias-free primitive operations of StyleGAN3 (Karras et al., 2021) to achieve translation equivariance, i.e., to enforce the synthesis network to have no preferred positions for the generated features.

StyleGAN-XL has a unique discriminator design where multiple discriminator heads operate on feature projections (Sauer et al., 2021) from two frozen, pretrained feature extraction networks: DeiT-M (Touvron et al., 2021a) and EfficientNet (Tan & Le, 2019). Their outputs are fed through randomized cross-channel and cross-scale mixing modules. This results in two feature pyramids with four resolution levels each that are then processed by eight discriminator heads. An additional pretrained classifier network is used to provide guidance during training.

	Zero-shot FID_{30k} ↓	CLIP score ↑
StyleGAN-XL	51.88	5.58
+ StyleGAN2 layers	51.60	5.61
+ Residual Convolutions	45.23	5.70
+ Split operation	45.10	6.02
+ Redesigned D	26.77	9.78
+ $\mathcal{L}_{\text{CLIP}}$	**20.52**	**11.72**

Table 1. **Architecture ablation.** Our architectural changes notably improve sample quality and text alignment. Here, we use the lightweight training configuration described in Appendix A.

The synthesis network of StyleGAN-XL is trained progressively, increasing the output resolution over time by introducing new synthesis layers once the current resolution stops improving. In contrast to a previous progressive growing approach (Karras et al., 2018), the discriminator structure does not change during training. Instead, the early low-resolution images are upsampled as necessary to suit the discriminator. In addition, the already trained synthesis layers are frozen as further layers are added.

For class-conditional synthesis, StyleGAN-XL concatenates an embedding of a one-hot class label to $\mathbf{z}$ and uses a projection discriminator (Miyato & Koyama, 2018).

3. StyleGAN-T

We choose StyleGAN-XL as our baseline architecture because of its strong performance in class-conditional ImageNet synthesis (Sauer et al., 2022). In this section, we modify this baseline piece by piece, focusing on the generator (Section 3.1), discriminator (Section 3.2), and variation vs. text alignment tradeoff mechanisms (Section 3.3) in turn.

Throughout the redesign process, we measure the effect of our changes using zero-shot MS COCO. For practical reasons, the tests use a limited compute budget, smaller models, and a smaller dataset than the large-scale experiments in Section 4; see Appendix A for details. We quantify sample quality using FID (Heusel et al., 2017) and text alignment using CLIP score (Hessel et al., 2021). Following prior art (Balaji et al., 2022), we compute the CLIP score using a ViT-g-14 model trained on LAION-2B (Schuhmann et al., 2022).

To change the class conditioning to text conditioning in our baseline model, we embed the text prompts using a pretrained CLIP ViT-L/14 text encoder (Radford et al., 2021) and use them in place of the class embedding. Accordingly, we also remove the training-time classifier guidance. This simple conditioning mechanism matches the early text-to-

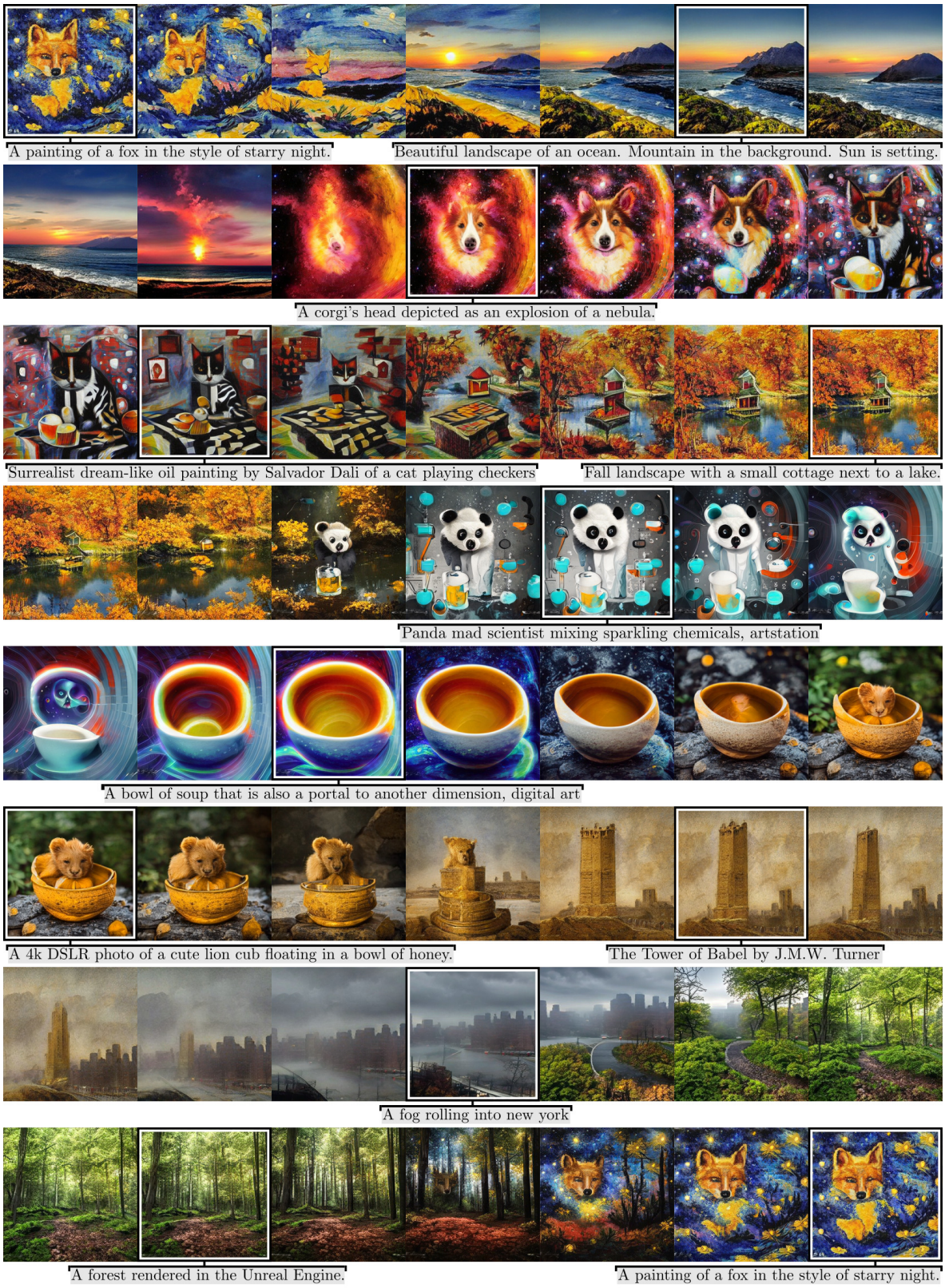

Figure 2. **Example images and interpolations.** StyleGAN-T generates diverse samples matching the text prompt and allows for smooth interpolations between prompts, illustrated as a single continuous interpolation in scanline order. Generating these 56 samples at 512×512 takes 6 seconds on an NVIDIA RTX 3090, while a comparable grid takes up to several minutes with current diffusion models. The accompanying video further demonstrates interpolations and contrasts them with diffusion models.

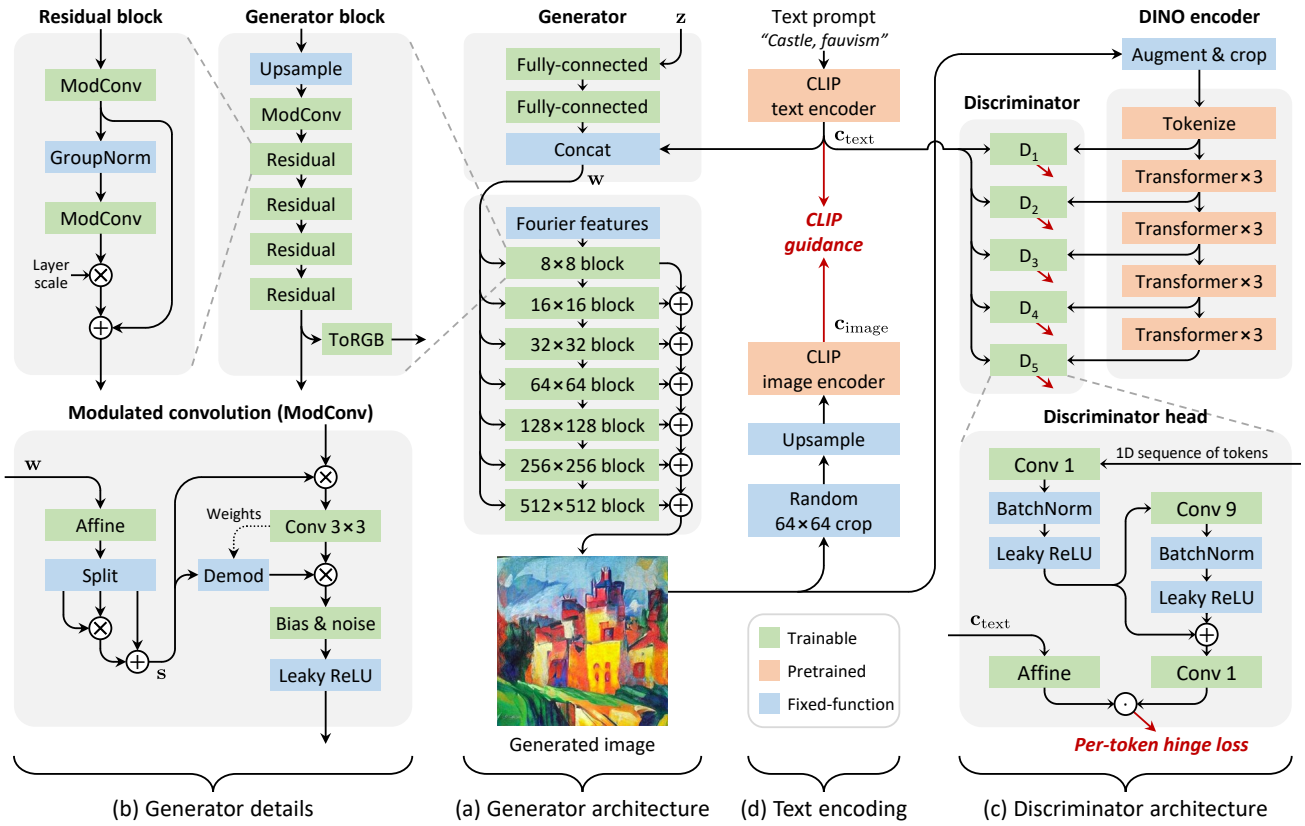

Figure 3. **Overview of StyleGAN-T. (a)** Our generator architecture (Sec. 3.1) is closely related to StyleGAN2, with the learned constant replaced with Fourier features and conditioning applied in a slightly different place. **(b)** For each resolution, a generator block is executed and its contribution is accumulated to the image via a dedicated ToRGB layer. The generator blocks employ residual connections and a new 2nd order style mechanism (Eq. 1). **(c)** Our discriminator (Sec. 3.2) processes the intermediate tokens of a DINO-trained vision transformer using 5 identical discriminator heads. Text conditioning is done using projection at the end. **(d)** Text prompt is embedded using CLIP and supplied to the generator and discriminator. We also employ a guidance term to further improve text alignment (Sec. 3.3).

image models (Reed et al., 2016a;b). Throughout this work, we use CLIP ViT-L/14 for text conditioning as it offers a good trade-off between synthesis performance and computational overhead, compared to using larger language models (Saharia et al., 2022). As shown in Table 1, this baseline reaches a zero-shot FID of 51.88 and CLIP score of 5.58 in our lightweight training configuration. Note that we use a different CLIP model for conditioning the generator and for computing the CLIP score, which reduces the risk of artificially inflating the results.

3.1. Redesigning the Generator

StyleGAN-XL uses StyleGAN3 layers to achieve translational equivariance. While equivariance can be desirable for various applications, we do not expect it to be necessary for text-to-image synthesis because none of the successful DM/ARM-based methods are equivariant. Additionally, the equivariance constraint adds computational cost and poses certain limitations to the training data that large-scale image datasets typically violate (Karras et al., 2021).

For these reasons, we drop the equivariance and switch to StyleGAN2 backbone for the synthesis layers, including output skip connections and spatial noise inputs that facilitate stochastic variation of low-level details. The high-level architecture of our generator after these changes is shown in Fig. 3a. We additionally propose two changes to the details of the generator architecture (Fig. 3b).

Residual convolutions. As we aim to increase the model capacity significantly, the generator must be able to scale in both width and depth. However, in the basic configuration, a significant increase in the generator's depth leads to an early mode collapse in training. An important building block in modern CNN architectures (Liu et al., 2022b; Dhariwal & Nichol, 2021) is an easily optimizable residual block that normalizes the input and scales the output. Following these insights, we make half the convolution layers residual and wrap them by GroupNorm (Wu & He, 2018) for normalization and Layer Scale (Touvron et al., 2021b) for scaling their contribution. A layer scale of a low initial value of 10^{-5} allows gradually fading in the convolution layer's contri-

bution, stabilizing the early training iterations significantly. This design allows us to increase the total number of layers considerably — by approximately 2.3× in the lightweight configuration and 4.5× in the final model. For fairness, we match the parameter count of the StyleGAN-XL baseline.

Stronger conditioning. The text-to-image setting is challenging because the factors of variation can vastly differ per prompt. Consider the prompts "a close-up of a face" and "a beautiful landscape." The first prompt should generate faces with varying eye color, skin color, and proportions, whereas the second should produce landscapes from different areas, seasons, and daytime. In a style-based architecture, all of this variation has to be implemented by the per-layer styles. Thus the text conditioning may need to affect the styles much more strongly than was necessary for simpler settings.

In early tests, we observed a clear tendency of the input latent $\mathbf{z}$ to dominate over the text embedding $\mathbf{c}_{\text{text}}$ in our baseline architecture, leading to poor text alignment. To remedy this, we introduce two changes that aim to amplify the role of $\mathbf{c}_{\text{text}}$. First, we let the text embeddings bypass the mapping network, following the observations by Härkönen et al. (2022). A similar design was also used in LAFITE (Zhou et al., 2022), assuming that the CLIP text encoder defines an appropriate intermediate latent space for the text conditioning. We thus concatenate $\mathbf{c}_{\text{text}}$ directly to $\mathbf{w}$ and use a set of affine transforms to produce per-layer styles $\tilde{\mathbf{s}}$. Second, instead of using the resulting $\tilde{\mathbf{s}}$ to modulate the convolutions as-is, we further split it into three vectors of equal dimension $\tilde{\mathbf{s}}_{1,2,3}$ and compute the final style vector as

$$\mathbf{s} = \tilde{\mathbf{s}}_1 \odot \tilde{\mathbf{s}}_2 + \tilde{\mathbf{s}}_3. \tag{1}$$

The crux of this operation is the element-wise multiplication $\odot$ that effectively turns the affine transform into a 2nd order polynomial network (Chrysos et al., 2020; Chrysos & Panagakis, 2021), increasing its expressive power. The stacked MLP-based conditioning layers in DF-GAN (Tao et al., 2022) implicitly include similar 2nd order terms.

Together, our changes to the generator improve FID and CLIP score by ∼10%, as shown in Table 1.

3.2. Redesigning the Discriminator

We redesign the discriminator from scratch but retain StyleGAN-XL's key ideas of relying on a frozen, pretrained feature network and using multiple discriminator heads.

Feature network. For the feature network, we choose a ViT-S (Dosovitskiy et al., 2021) trained with the self-supervised DINO objective (Caron et al., 2021). The network is lightweight, fast to evaluate, and encodes semantic information at high spatial resolution (Amir et al., 2021). An additional benefit of using a self-supervised feature network is that it circumvents the concern of potentially compromising FID (Kynkäänniemi et al., 2022).

Architecture. Our discriminator architecture is shown in Fig. 3c. ViTs are isotropic, i.e., the representation size (tokens × channels) and receptive field (global) are the same throughout the network. This isotropy allows us to use the same architecture for all discriminator heads, which we space equally between the transformer layers. Multiple heads are known to be beneficial (Sauer et al., 2021), and we use five heads in our design.

Our discriminator heads are minimalistic, as detailed in Fig. 3c, bottom. The residual convolution's kernel width controls the head's receptive field in the token sequence. We found that 1D convolutions applied on the sequence of tokens performed just as well as 2D convolutions applied on spatially reshaped tokens, indicating that the discrimination task does not benefit from whatever 2D structure remains in the tokens. We evaluate a hinge loss (Lim & Ye, 2017) independently for each token in every head.

Sauer et al. (2021) use synchronous BatchNorm (Ioffe & Szegedy, 2015) to provide batch statistics to the discriminator. BatchNorm is problematic when scaling to a multi-node setup, as it requires communication between nodes and GPUs. We use a variant that computes batch statistics on small virtual batches (Hoffer et al., 2017). The batch statistics are not synchronized between devices but are calculated per local minibatch. Furthermore, we do not use running statistics, and thus no additional communication overhead between GPUs is introduced.

Augmentation. We apply differentiable data augmentation (Zhao et al., 2020) with default parameters before the feature network in the discriminator. We use random crops when training at a resolution larger than 224×224 pixels (ViT-S training resolution).

As shown in Table 1, these changes significantly improve FID and CLIP score by further ∼40%. This considerable improvement indicates that a well-designed discriminator is critical when dealing with highly diverse datasets. Compared to the StyleGAN-XL discriminator, our simplified redesign is ∼2.5× faster, leading to ∼1.5× faster training.

3.3. Variation vs. Text Alignment Tradeoffs

Guidance (Dhariwal & Nichol, 2021; Ho & Salimans, 2022) is an essential component of current text-to-image diffusion models. It trades variation for perceived image quality in a principled way, preferring images that are strongly aligned with the text conditioning. In practice, guidance drastically improves the results; thus, we want to approximate its behavior in the context of GANs.

Guiding the generator. StyleGAN-XL uses a pretrained ImageNet classifier to provide additional gradients during training, guiding the generator toward images that are easy to classify. This method improves results significantly. In the context of text-to-image, "classification" involves captioning the images. Thus, a natural extension of this approach is to use a CLIP image encoder instead of a classifier. Following Crowson et al. (2022), at each generator update, we pass the generated image through the CLIP image encoder to obtain caption $\mathbf{c}_{\text{image}}$, and minimize the squared spherical distance to the normalized text embedding $\mathbf{c}_{\text{text}}$:

$$\mathcal{L}_{\text{CLIP}} = \arccos^2(\mathbf{c}_{\text{image}} \cdot \mathbf{c}_{\text{text}}) \tag{2}$$

This additional loss term guides the generated distribution towards images that are captioned similarly to the input text encoding $\mathbf{c}_{\text{text}}$. Its effect is thus similar to the guidance in diffusion models. Fig. 3d illustrates our approach.

CLIP has been used in prior work to guide a pretrained generator during synthesis (Nichol et al., 2022; Crowson et al., 2022; Liu et al., 2021). In contrast, we use it as a part of the loss function during training. It is important to note that overly strong CLIP guidance during training impairs FID, as it limits the distribution diversity and ultimately starts introducing image artifacts. Therefore, the weight of $\mathcal{L}_{\text{CLIP}}$ in the overall loss needs to strike a balance between image quality, text conditioning, and distribution diversity; we set it to 0.2. We further observed that guidance is helpful only up to 64×64 pixel resolution. At higher resolutions, we apply $\mathcal{L}_{\text{CLIP}}$ to random 64×64 pixel crops.

As shown in Table 1, CLIP guidance improves FID and CLIP scores by further ∼20%.

Guiding the text encoder. Interestingly, the earlier methods listed above that use a pretrained generator did not report encountering low-level image artifacts. We hypothesize that the frozen generator acts as a prior that suppresses them. We build on this insight to further improve the text alignment. In our primary training phase, the generator is trainable and the text encoder is frozen. We then introduce a secondary phase, where the generator is frozen and the text encoder becomes trainable instead. We only train the text encoder as far as the generator conditioning is concerned; the discriminator and the guidance term (Eq. 2) still receive $\mathbf{c}_{\text{text}}$ from the original frozen encoder. This secondary phase allows a very high CLIP guidance weight of 50 without introducing artifacts and significantly improves text alignment without compromising FID (Section 4.2). Compared to the primary phase, the secondary phase can be much shorter. After convergence, we continue with the primary phase.

Explicit truncation. Typically variation has been traded to higher fidelity in GANs using the truncation trick (Marchesi, 2017; Brock et al., 2019; Karras et al., 2019), where a sampled latent $\mathbf{w}$ is interpolated towards its mean with respect to the given conditioning input. This way, truncation pushes $\mathbf{w}$ to a higher-density region where the model performs better. In our implementation, $\mathbf{w} = [f(\mathbf{z}),\ \mathbf{c}_{\text{text}}]$, where $f(\cdot)$ denotes the mapping network, so the per-prompt mean is given by $\tilde{\mathbf{w}} = \mathbb{E}_{\mathbf{z}}[\mathbf{w}] = [\tilde{\mathbf{f}},\ \mathbf{c}_{\text{text}}]$, where $\tilde{\mathbf{f}} = \mathbb{E}_{\mathbf{z}}[f(\mathbf{z})]$. We thus implement truncation by tracking $\tilde{\mathbf{f}}$ during training and interpolating between $\tilde{\mathbf{w}}$ and $\mathbf{w}$ according to scaling parameter $\psi \in [0, 1]$ at inference time.

Figure 4. **Truncation.** Four samples for the prompt "a graphite sketch of Eva Longoria" with different random $\mathbf{z}$. Increasing truncation (decreasing ψ) improves the text alignment according to mean CLIP score per row ($\overline{CS}$) at the cost of lower variation.

We illustrate the impact of truncation in Fig. 4. In practice, we rely on the combination of CLIP guidance and truncation. Guidance improves the model's overall text alignment, and truncation can further boost quality and alignment for a given sample, trading away some variation.

4. Experiments

Using the final configuration developed in Section 3, we scale the model size, dataset, and training time. Our final model consists of ∼1 billion parameters; we did not observe any instabilities when increasing the model size. We train on a union of several datasets amounting to 250M text-image pairs in total. We use progressive growing similar to StyleGAN-XL, except that all layers remain trainable. The hyperparameters and dataset details are listed in Appendix A.

The total training time was four weeks on 64 A100 GPUs using a batch size of 2048. We first trained the primary phase for 3 weeks (resolutions up to 64×64), then the secondary phase for 2 days (text embedding), and finally the primary phase again for 5 days (resolutions up to 512×512). For comparison, our total compute budget is about a quarter of Stable Diffusion's (CompVis, 2022).

Model	Model type	Zero-shot FID_{30k}	Speed [s]
Stable Diffusion *	Diffusion	8.40	–
eDiff-I	Diffusion	7.60	26.0
LDM *	Diffusion	7.59	–
GLIDE	Diffusion	7.40	10.9
LAFITE *	GAN	14.80	∼**0.01**
StyleGAN-T	GAN	**7.30**	0.06

* downsampled to 64×64 pixels using Lanczos – not available

Table 2. **Comparison of FID on MS COCO 64×64.** Inference speeds are measured on an A100. For LAFITE we estimate what its speed would be at a native 64×64 resolution.

Model	Model type	Zero-shot FID_{30k}	Speed [s]
LDM	Diffusion	12.63	3.7
GLIDE	Diffusion	12.24	15.0
DALL·E 2	Diffusion	10.39	–
Stable Diffusion *	Diffusion	8.59	3.7
Imagen	Diffusion	7.27	9.1
eDiff-I	Diffusion	**6.95**	32.0
DALL·E	Autoregressive	27.50	–
Ernie-ViLG	Autoregressive	14.70	–
Make-A-Scene *	Autoregressive	11.84	25.0
Parti-3B	Autoregressive	8.10	6.4
Parti-20B	Autoregressive	7.23	–
LAFITE	GAN	26.94	**0.02**
StyleGAN-T *	GAN	13.90	0.10

* downsampled to 256×256 pixels using Lanczos – not available

Table 3. **Comparison of FID on MS COCO 256×256.** Inference speeds are measured on an A100, except for Imagen and Parti that use a faster TPUv4 accelerator. The Stable Diffusion numbers are from (Balaji et al., 2022; Lambda Labs, 2022); the other numbers are obtained from the respective papers or through correspondence with the authors.

4.1. Quantitative Comparison to State-of-the-Art

We use zero-shot MS COCO to compare the performance of our model to the state-of-the-art quantitatively at 64×64 pixel output resolution in Table 2 and 256×256 in Table 3. At low resolution, StyleGAN-T outperforms all other approaches in terms of output quality, while being very fast to evaluate. In this test we use the model before the final training phase, i.e., one that produces 64×64 images natively. At high resolution, StyleGAN-T still significantly outperforms LAFITE but lags behind DMs and ARMs in terms of FID.

These results lead us to two conclusions. First, GANs can match or even beat current DMs in large-scale text-to-image synthesis at low resolution. Second, a powerful superresolution model is crucial. While FID slightly decreases in eDiff-I when moving from 64×64 to 256×256 (7.60→6.95), it currently almost doubles in StyleGAN-T. Therefore, it is evident that StyleGAN-T's superresolution stage is underperforming, causing a gap to the current state-of-the-art high-resolution results. Whether this gap can be bridged simply with additional capacity or longer training is an open question.

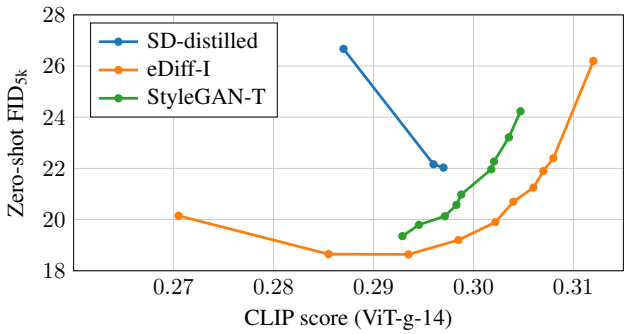

Figure 5. **Comparing text alignment tradeoffs.** We compare FID–CLIP score curves of StyleGAN-T, distilled Stable Diffusion (SD-distilled), and eDiff-I. We report values of SD-distilled at a guidance scale of $w = 4$. For a fair comparison, we report numbers for CLIP-conditioned eDiff-I disabling additional conditioning on T5-XXL text embeddings. The models use different methods to increase the CLIP score (i.e., text alignment): StyleGAN-T decreases truncation $\psi = \{1.0 \ldots 0.0\}$, SD-distilled increases the number of sampling steps $\{2, 4, 8\}$, eDiff-I increases guidance scale $w = \{0 \ldots 10\}$.

4.2. Evaluating Variation vs. Text Alignment

We report FID–CLIP score curves in Fig. 5. We compare StyleGAN-T to a strong DM baseline (CLIP-conditioned variant of eDiff-I) and a fast, distilled DM baseline (SD-distilled) (Meng et al., 2022).

Using Truncation, StyleGAN-T can push the CLIP score to 0.305, successfully improving text alignment. StyleGAN-T outperforms SD-distilled in both FID and CLIP scores yet remains behind eDiff-I. Regarding speed, eDiff-I requires 32.0 seconds to generate a sample. SD-distilled is significantly faster and only needs 0.6 seconds at its best performance at eight sampling steps. StyleGAN-T beats both baselines, generating a sample in 0.1 seconds.

To isolate the impact of text encoder training, we evaluate FID–CLIP score curves in Fig. 6. For this experiment, we utilize the same generator network and only swap the text encoder. As the generator has been frozen in the secondary phase, it can handle both the original and fine-tuned CLIP text embeddings as evidenced by their equal performance measured by FID. Fine-tuning the text encoder significantly improves the CLIP score without compromising FID.

4.3. Qualitative Results

Fig. 2 shows example images produced by StyleGAN-T, along with interpolations between them. The accompanying video shows this in animation and compares it to diffusion models, demonstrating that the interpolation properties of GANs continue to be considerably smoother. We provide

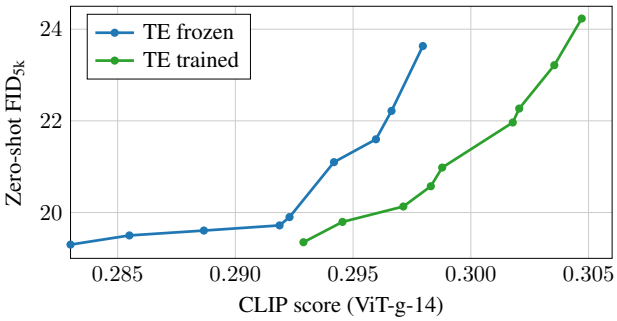

Figure 6. **Text encoder training.** Training the CLIP text encoder (TE) pushes the entire FID–CLIP score curve to the right, hence, increasing overall text alignment.

Figure 7. **Latent manipulation.** Samples (first column) can be manipulated by following semantic directions in latent space.

more qualitative examples in Appendix B.

Interpolating between different text prompts is straightforward. For an image generated by an intermediate latent $\mathbf{w}_0 = [f(\mathbf{z}),\ \mathbf{c}_{\text{text0}}]$, we substitute the text condition $\mathbf{c}_{\text{text0}}$ with a new text condition $\mathbf{c}_{\text{text1}}$. We then interpolate $\mathbf{w}_0$ towards the new latent $\mathbf{w}_1 = [f(\mathbf{z}),\ \mathbf{c}_{\text{text1}}]$ as shown in Fig. 7. This approach is similar to DALL·E 2's text diff operation that interpolates between CLIP embeddings. Previous work for manipulating GAN-generated images (Patashnik et al., 2021) typically discovers these latent directions via a training process that needs to be repeated per prompt and is, therefore, expensive. Meaningful latent directions are a built-in property of our model, and no extra training is needed.

By appending different styles to a prompt, StyleGAN-T can generate a wide variety of styles as shown in Fig. 8. Subjects tend to be aligned for a fixed latent $\mathbf{z}$, which we showcase in the accompanying video.

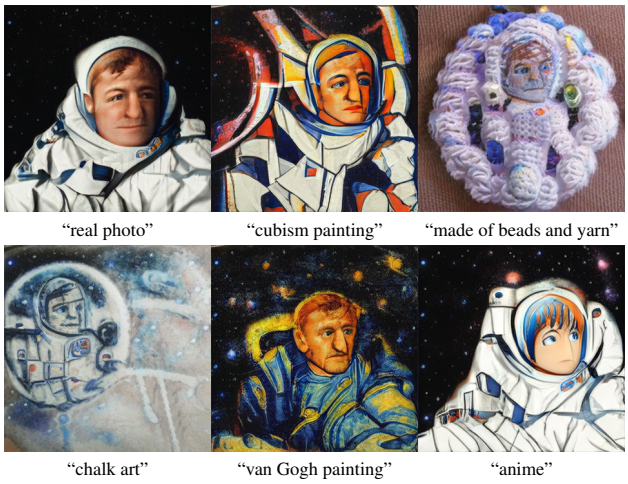

Figure 8. **Styles.** Samples generated by StyleGAN-T for a fixed random seed and the caption "astronaut, {X}", where X is denoted below each image.

Figure 9. **Failure cases.** StyleGAN-T can struggle to bind attributes to objects, and to produce coherent text.

5. Limitations and Future Work

Similarly to DALL·E 2 that also uses CLIP as the underlying language model, StyleGAN-T sometimes struggles in terms of binding attributes to objects as well as producing coherent text in images (Fig. 9). Using a larger language model would likely resolve this issue at the cost of slower runtime (Saharia et al., 2022; Balaji et al., 2022).

Guidance via CLIP loss is vital for good text alignment, but high guidance strength results in image artifacts. A possible solution could be to retrain CLIP on higher-resolution data that does not suffer from aliasing or other image quality issues. In this context, the conditioning mechanism in the discriminator may also be worth revisiting.

Truncation improves text alignment but differs from guidance in diffusion models in two important ways. While truncation is always towards a single mode, guidance can at least theoretically be arbitrarily multi-modal. Also, truncation sharpens the distribution before the synthesis network, which can reshape the distribution in arbitrary ways, thus, possibly undoing any prior sharpening. Therefore, alternative methods to truncation might further improve the results.

Improved super-resolution stages (i.e., high-resolution lay-

ers) through higher capacity and longer training are an obvious avenue for future work. Concurrent work (Kang et al., 2023) indicates that GAN-based upsamplers can outperform DMs in a large-scale setting, given enough capacity and compute. Training directly in a pretrained token space, delegating superresolution to a pretrained decoder, presents another potential route. Recent evidence from MUSE (Chang et al., 2023) supports its effectiveness for fast text-to-image synthesis.

Methods for "personalizing" diffusion models have become popular (Ruiz et al., 2022; Gal et al., 2022). They finetune a pretrained model to associate a unique identifier with a given subject, allowing it to synthesize novel images of the same subject in novel contexts. Such approaches can be similarly applied to GANs (Nitzan et al., 2022).

Acknowledgements

We would like to thank Tim Brooks, Miika Aittala, and Jaakko Lehtinen for helpful discussions; Yogesh Balaji and Seungjun Nah for computing additional metrics for eDiff-I; Tuomas Kynkäänniemi and Koki Nagano for feedback on an earlier draft; Tero Kuosmanen, Samuel Klenberg, and Janne Hellsten for maintaining the compute infrastructure; and David Luebke and Vanessa Sauer for their general support.

References

Abdal, R., Zhu, P., Mitra, N. J., and Wonka, P. StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows. *ACM Trans. Graph.*, 40(3), 2021. 1

Amir, S., Gandelsman, Y., Bagon, S., and Dekel, T. Deep VIT features as dense visual descriptors. *CoRR*, abs/2112.05814, 2021. 5

Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *CoRR*, abs/2211.01324, 2022. 2, 7, 8

Bermano, A. H., Gal, R., Alaluf, Y., Mokady, R., Nitzan, Y., Tov, O., Patashnik, O., and Cohen-Or, D. State-of-the-art in the architecture, methods and applications of StyleGAN. *CoRR*, abs/2202.14020, 2022. 1

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *Proc. ICLR*, 2019. 6

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proc. ICCV*, 2021. 5

Chang, H., Zhang, H., Barber, J., Maschinot, A., Lezama, J., Jiang, L., Yang, M.-H., Murphy, K., Freeman, W. T., Rubinstein, M., et al. Muse: Text-to-image generation via masked generative transformers. *CoRR*, 2023. 9

Changpinyo, S., Sharma, P., Ding, N., and Soricut, R. Conceptual 12M: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. In *Proc. CVPR*, 2021. 12

Chrysos, G., Moschoglou, S., Bouritsas, G., Deng, J., Panagakis, Y., and Zafeiriou, S. Deep polynomial neural networks. In *Proc. CVPR*, 2020. 5

Chrysos, G. G. and Panagakis, Y. CoPE: conditional image generation using polynomial expansions. In *Proc. NeurIPS*, 2021. 5

CompVis. Stable diffusion model card, 2022. URL https://github.com/CompVis/stable-diffusion/blob/main/Stable_Diffusion_v1_Model_Card.md. 2, 6

Crowson, K., Biderman, S., Kornis, D., Stander, D., Hallahan, E., Castricato, L., and Raff, E. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. In *Proc. ECCV*, 2022. 6

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 2

Desai, K., Kaul, G., Aysola, Z., and Johnson, J. RedCaps: web-curated image-text data created by the people, for the people. In *Proc. Neurips (Datasets and Benchmarks)*, 2021. 12

Dhariwal, P. and Nichol, A. Diffusion models beat GANs on image synthesis. In *Proc. NeurIPS*, 2021. 2, 4, 5

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 5

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *Proc. ECCV*, 2022. 1

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *CoRR*, abs/2208.01618, 2022. 9

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. In *Proc. NIPS*, 2014. 1

Härkönen, E., Hertzmann, A., Lehtinen, J., and Paris, S. GANSpace: Discovering interpretable GAN controls. In *Proc. NeurIPS*, 2020. 1

Härkönen, E., Aittala, M., Kynkäänniemi, T., Laine, S., Aila, T., and Lehtinen, J. Disentangling random and cyclic effects in time-lapse sequences. *ACM Trans. Graph.*, 41 (4), 2022. 5

Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. CLIPScore: A reference-free evaluation metric for image captioning. In *Proc. EMNLP*, 2021. 2

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Proc. NeurIPS*, 2017. 2

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *CoRR*, abs/2207.12598, 2022. 5

Hoffer, E., Hubara, I., and Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Proc. NeurIPS*, 2017. 5

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. ICML*, 2015. 5

Kafri, O., Patashnik, O., Alaluf, Y., and Cohen-Or, D. StyleFusion: A generative model for disentangling spatial segments. *ACM Trans. Graph.*, 41(5):1–15, 2022. 1

Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., and Park, T. Scaling up gans for text-to-image synthesis. *Proc. CVPR*, 2023. 9

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *Proc. ICLR*, 2018. 2

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019. 1, 2, 6

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of StyleGAN. In *Proc. CVPR*, 2020. 1, 2

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021. 1, 2, 4

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022. 1

Kynkäänniemi, T., Karras, T., Aittala, M., Aila, T., and Lehtinen, J. The role of ImageNet classes in Fréchet inception distance. *CoRR*, abs/2203.06026, 2022. 5

Lambda Labs. All you need is one gpu: Inference benchmark for stable diffusion, 2022. URL https://lambdalabs.com/blog/inference-benchmark-stable-diffusion. 7

Lim, J. H. and Ye, J. C. Geometric Gan. *CoRR*, 2017. 5

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014. 2

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. *Proc. ICLR*, 2022a. 14

Liu, X., Gong, C., Wu, L., Zhang, S., Su, H., and Liu, Q. FuseDream: Training-free text-to-image generation with improved clip+ gan space optimization. *CoRR*, abs/2112.01573, 2021. 6

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *Proc. CVPR*, 2022b. 4

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. DPM-Solver: A fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In *Proc. NeurIPS*, 2022. 1

Marchesi, M. Megapixel size image creation using generative adversarial networks. *CoRR*, abs/1706.00082, 2017. 6

Meng, C., Gao, R., Kingma, D. P., Ermon, S., Ho, J., and Salimans, T. On distillation of guided diffusion models. *CoRR*, abs/2210.03142, 2022. 7

Miyato, T. and Koyama, M. cGANs with projection discriminator. In *Proc. ICLR*, 2018. 2

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proc. ICML*, 2022. 6

Nitzan, Y., Aberman, K., He, Q., Liba, O., Yarom, M., Gandelsman, Y., Mosseri, I., Pritch, Y., and Cohen-Or, D. Mystyle: A personalized generative prior. *ACM Trans. Graph.*, 2022. 9

OpenAI. Dall·e api, 2022. URL https://openai.com/product/dall-e-2. 14

Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D. StyleCLIP: Text-driven manipulation of StyleGAN imagery. In *Proc. ICCV*, 2021. 8

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. 2

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. 1, 12

Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., and Lee, H. Learning what and where to draw. In *Proc. NeurIPS*, 2016a. 4

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. In *Proc. ICML*, 2016b. 4

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, 2022. 1, 12

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *CoRR*, abs/2208.12242, 2022. 9

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Proc. NeurIPS*, 2022. 1, 4, 8

Salimans, T. and Ho, J. Progressive distillation for fast sampling of diffusion models. *CoRR*, abs/2202.00512, 2022. 1

Sauer, A., Chitta, K., Müller, J., and Geiger, A. Projected GANs converge faster. In *Proc. NeurIPS*, 2021. 2, 5

Sauer, A., Schwarz, K., and Geiger, A. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *Proc. SIGGRAPH*, 2022. 2

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. LAION-5B: An open large-scale dataset for training next generation image-text models. In *Proc. NeurIPS*, 2022. 1, 2, 12

Sharma, P., Ding, N., Goodman, S., and Soricut, R. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proc. ACL*, volume 1, pp. 2556–2565, 2018. 12

Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the latent space of GANs for semantic face editing. In *Proc. CVPR*, 2020. 1

Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., and Kiela, D. Flava: A foundational language and vision alignment model. In *Proc. CVPR*, 2022. 12

Tan, M. and Le, Q. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proc. ICML*, 2019. 2

Tao, M., Tang, H., Wu, F., Jing, X.-Y., Bao, B.-K., and Xu, C. DF-GAN: A simple and effective baseline for text-to-image synthesis. In *Proc. CVPR*, 2022. 5

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., and Li, L.-J. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 12

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In *Proc. ICML*, 2021a. 2

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *Proc. ICCV*, 2021b. 4

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., and Change Loy, C. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proc. ECCV*, 2018. 2

Wu, Y. and He, K. Group normalization. In *Proc. CVPR*, 2018. 4

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *CoRR*, abs/2206.10789, 2022. 1

Zhang, H., Yin, W., Fang, Y., Li, L., Duan, B., Wu, Z., Sun, Y., Tian, H., Wu, H., and Wang, H. ERNIE-ViLG: Unified generative pre-training for bidirectional vision-language generation. *CoRR*, abs/2112.15283, 2021. 1

Zhao, S., Liu, Z., Lin, J., Zhu, J.-Y., and Han, S. Differentiable augmentation for data-efficient GAN training. In *Proc. NeurIPS*, 2020. 5

Zhou, Y., Zhang, R., Chen, C., Li, C., Tensmeyer, C., Yu, T., Gu, J., Xu, J., and Sun, T. Towards language-free training for text-to-image generation. In *Proc. CVPR*, 2022. 1, 5

A. Configuration Details

Table 4 lists the training and network architecture hyperparameters for our two configurations: lightweight (used for ablations) and the full configuration (used for main results). Table 5 details the training schedules.

Lightweight training configuration. We train using the CC12M dataset (Changpinyo et al., 2021) at 64×64 resolution, without using progressive growing.

Full training configuration. We train using a union of several datasets: CC12m (Changpinyo et al., 2021), CC (Sharma et al., 2018), YFCC100m (filtered) (Thomee et al., 2016; Singh et al., 2022), Redcaps (Desai et al., 2021), LAION-aesthetic-6+ (Schuhmann et al., 2022). This amounts to a total of 250M text-image pairs. We use progressive growing similar to StyleGAN-XL, except that all layers remain trainable.

B. Qualitative Results

Fig. 10 shows additional examples of truncation. Fig. 11 shows qualitative comparisons to Latent Diffusion (Rombach et al., 2022), Stable Diffusion (Rombach et al., 2022), DALL·E 2 (Ramesh et al., 2022). We use the same prompts as in the truncation study.

	Lightweight	**Full**
Generator channel base	32768	65536
Generator channel max	512	2048
Number of residual blocks per generator block	3	4
Generator parameters	75 million	1.02 billion
Text encoder parameters	123 million	123 million
Latent (z) dimension	64	64
Discriminator's feature network	DINO ViT-S/16	DINO ViT-S/16
Discriminator head's input feature space size	384	384
Discriminator head's feature space size at text conditioning	64	64
Dataset size	12M	250M
Number of GPUs	8	64
Batch size	2048	2048
Optimizer	Adam	Adam
Generator learning rate	0.002	0.002
Generator Adam betas	(0, 0.99)	(0, 0.99)
Discriminator learning rate	0.002	0.002
Discriminator Adam betas	(0, 0.99)	(0, 0.99)
EMA	0.9978	0.9978
CLIP guidance weight	0.2	0.2 (primary phase), 50 (secondary phase)
Progressive growing	No	Yes

Table 4. Generator, discriminator, and training hyperparameters for the two setups used in this paper: Lightweight and Full configuration.

Lightweight	**Full**
Primary Phase	Primary Phase
64x64 for 50 A100 days (25 million iterations)	16x16 for 450 A100 days (118,000 iterations)
	32x32 for 450 A100 days (78,000 iterations)
	64x64 for 450 A100 days (57,000 iterations)
	Secondary Phase
	190 A100 days (20,000 iterations)
	Primary Phase
	128x128 for 96 A100 days (10,000 iterations)
	256x256 for 70 A100 days (6,000 iterations)
	512x512 for 30 A100 days (3,000 iterations)

Table 5. Training schedules for the two training configurations used in this paper. The times are listed as the number of days it would have taken on a single NVIDIA A100 GPU. An iteration corresponds to 2048 real and generated examples.

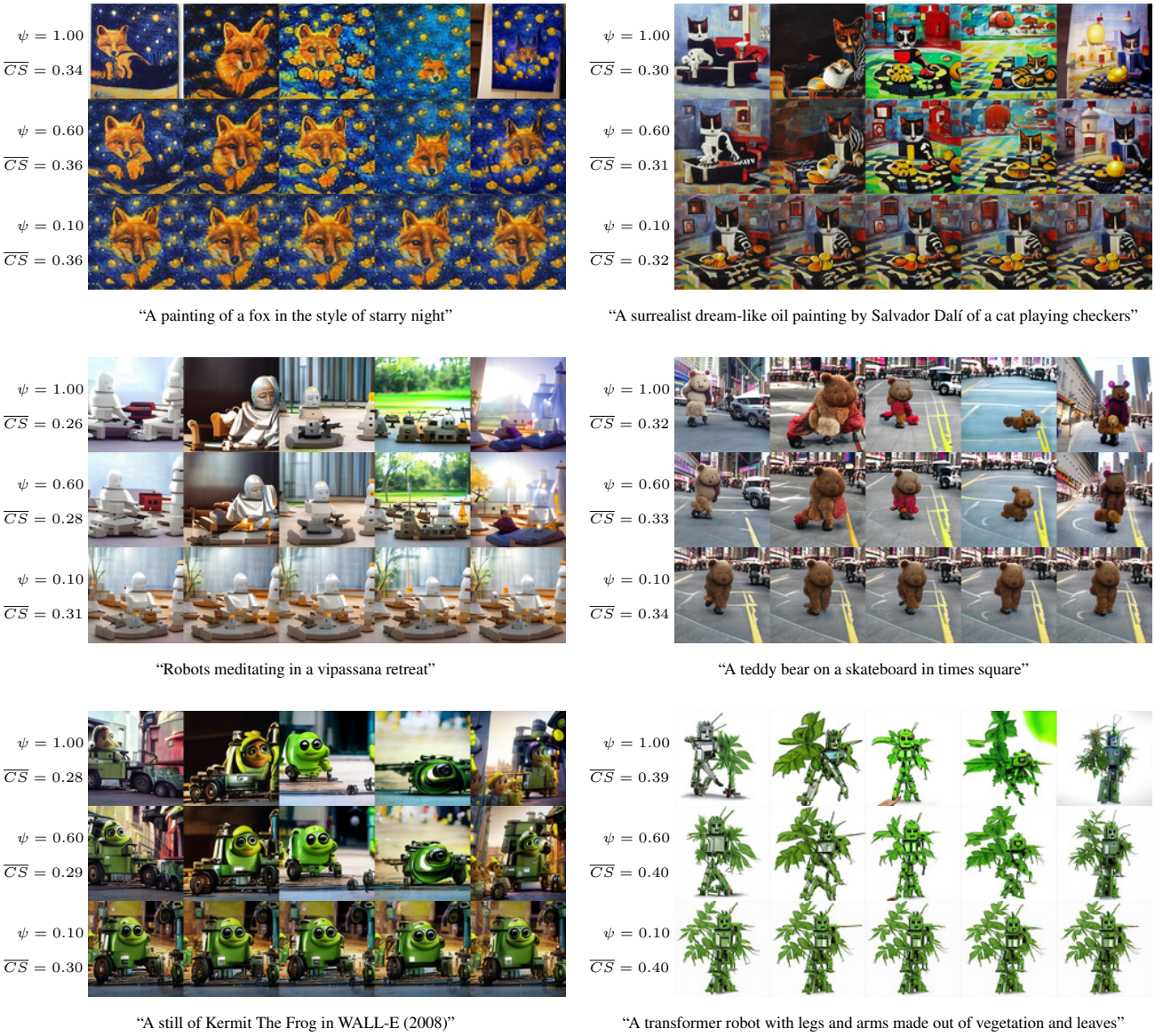

Figure 10. **Additional truncation grids.** We show samples for 6 different prompts and 5 different random latents, shared between the prompts. Increasing truncation (decreasing ψ), improves the text alignment according to mean CLIP score per row, $\overline{CS}$, at the cost of lower variation.

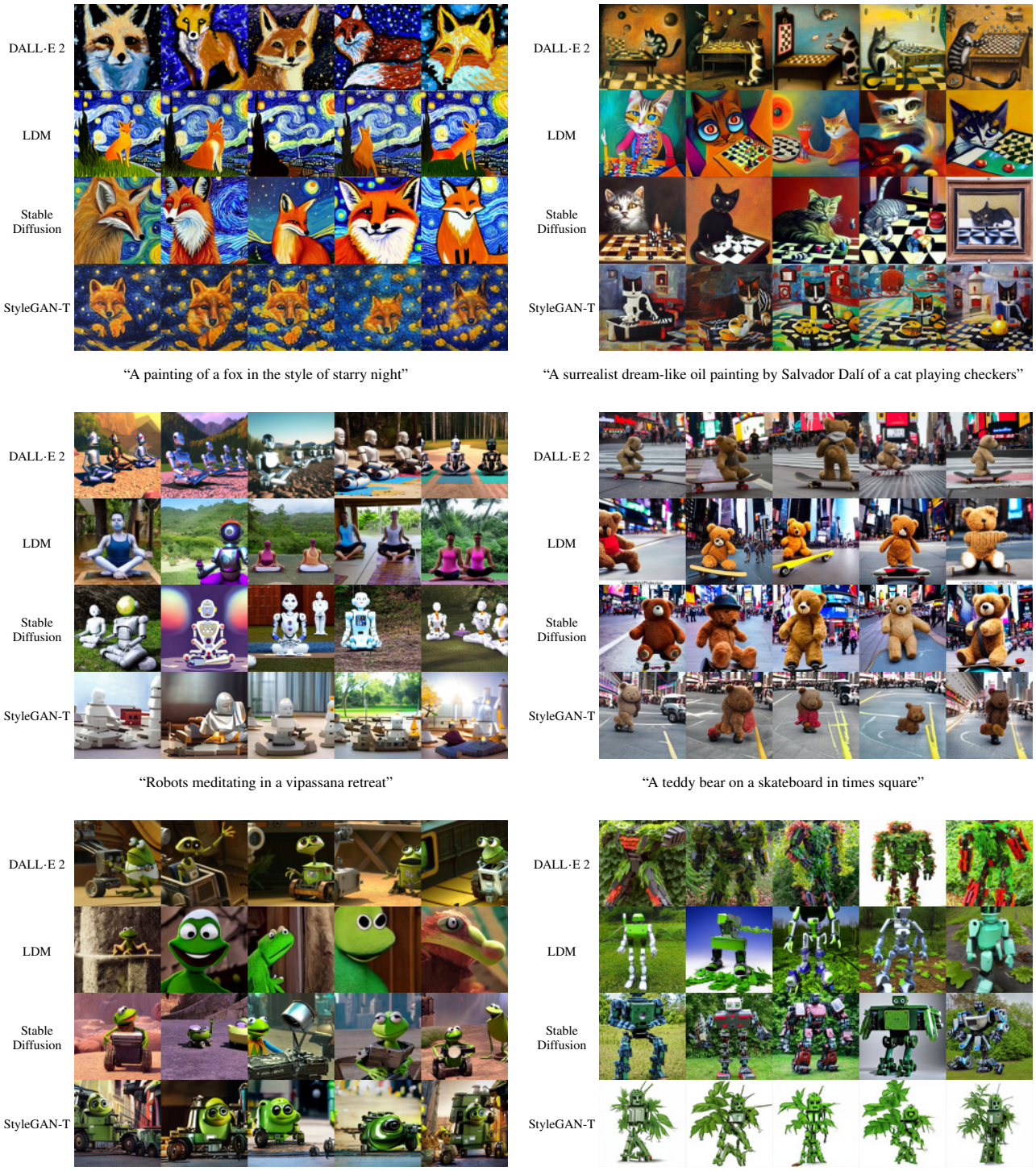

Figure 11. **Qualitative Comparisons.** We show samples for 6 different prompts and 5 different random latents, shared between the prompts. For StyleGAN-T, we set $\psi = 0.6$. LDM and Stable Diffusion utilize 250 and 50 sampling steps, respectively, utilizing the DDIM / PLMS sampler (Liu et al., 2022a). For DALL·E 2, we generate images via the official DALL·E service (OpenAI, 2022).