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

Can a single forward pass compete with diffusion? (ICML 2023)

Axel Sauer, Tero Karras, Samuli Laine, Andreas Geiger, Timo Aila

University of Tübingen · Tübingen AI Center · NVIDIA

Presented by: **Amogh Gupta**

The University of North Carolina at Chapel Hill

COMP 690 : Zhongzheng Ren

The Speed–Quality Tradeoff

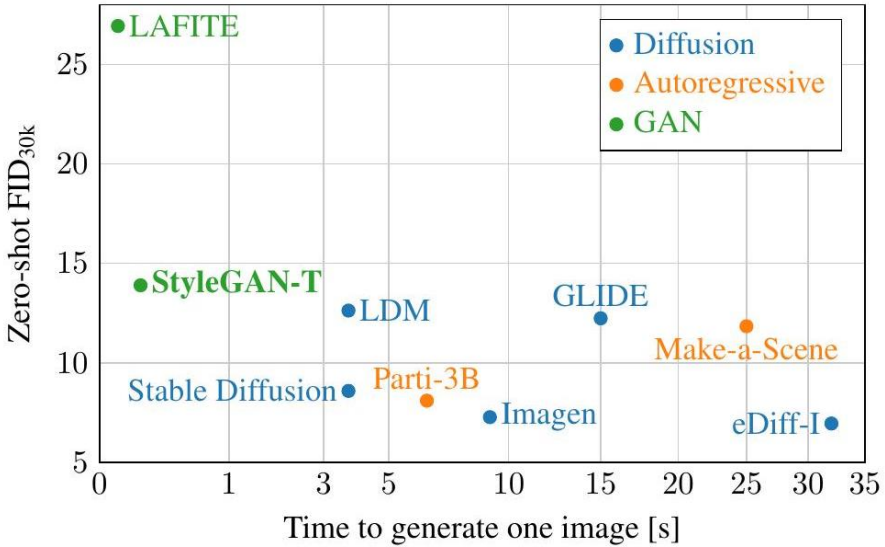


Figure 1: zero-shot FID on MS COCO 256×256 vs. time to generate one image

Diffusion and autoregressive models are iterative

A GAN needs one forward pass

But GANs were far behind on quality

What Made Text-to-Image Work

1. A large pretrained language model as the prompt encoder

The generator is not asked to learn language from image–caption pairs. It is handed embeddings from an encoder that already has general language understanding.

2. Web-scale image–caption training data

StyleGAN-T trains on a union of five datasets totalling 250M pairs.

Why Did GANs Fall Behind?

26.94
LAFITE zero-shot FID at 256×256

6.95
eDiff-I zero-shot FID at 256×256

~1B
StyleGAN-T parameters, trained stably

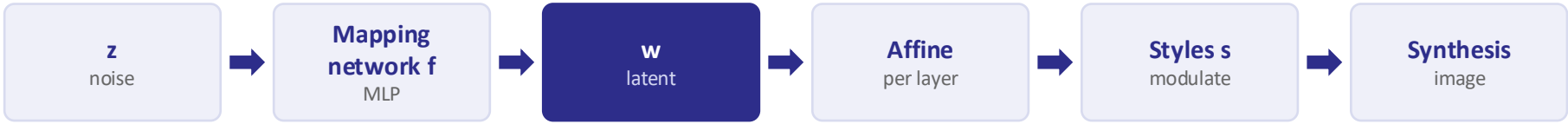
10 FPS
StyleGAN-T throughput on one A100

Diffusion and ARMs have scale and multi-modality built in

GANs get unstable as capacity grows

The precedent: the same thing happened on ImageNet

Recap: The StyleGAN Family



Mapping network → intermediate latent w

Styles: w never enters the network as a tensor

Weight demodulation (StyleGAN2)

StyleGAN-XL: The Starting Point

Generator: StyleGAN3 alias-free layers
Equivariant synthesis layers, so features have no preferred pixel position.

Discriminator: frozen pretrained features + many heads
Two frozen networks (DeiT-M, EfficientNet), cross-channel and cross-scale mixing.

Classifier guidance during training
A pretrained ImageNet classifier supplies extra gradients, pushing the generator toward images that are easy to classify.

Progressive growing, XL-style
New synthesis layers are added as resolution plateaus; the discriminator never changes shape, and trained layers are frozen.

Baseline after swapping the class label for a CLIP text embedding

FID 51.88
CLIP score 5.58

Converting class conditioning to text conditioning
The prompt is embedded with a frozen CLIP ViT-L/14 text encoder.

The Constraint: A Fixed Training Budget

≈ ¼

of Stable Diffusion's compute

64 × A100

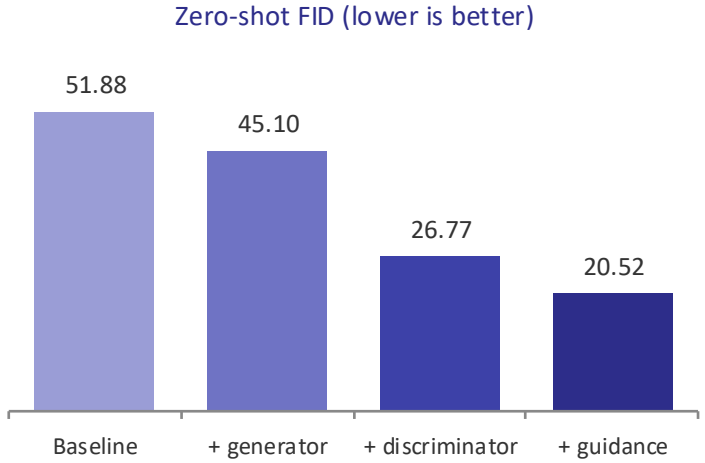
GPUs, batch size 2048

4 weeks

wall-clock training time

Architecture Ablation: What Actually Mattered

Configuration	FID ↓	CLIP ↑
StyleGAN-XL + CLIP text cond.	51.88	5.58
+ new generator	45.10	6.02
+ new discriminator	**26.77**	**9.78**
+ CLIP guidance loss	20.52	11.72



The discriminator redesign is worth more than everything else combined

StyleGAN-T: Architecture Overview

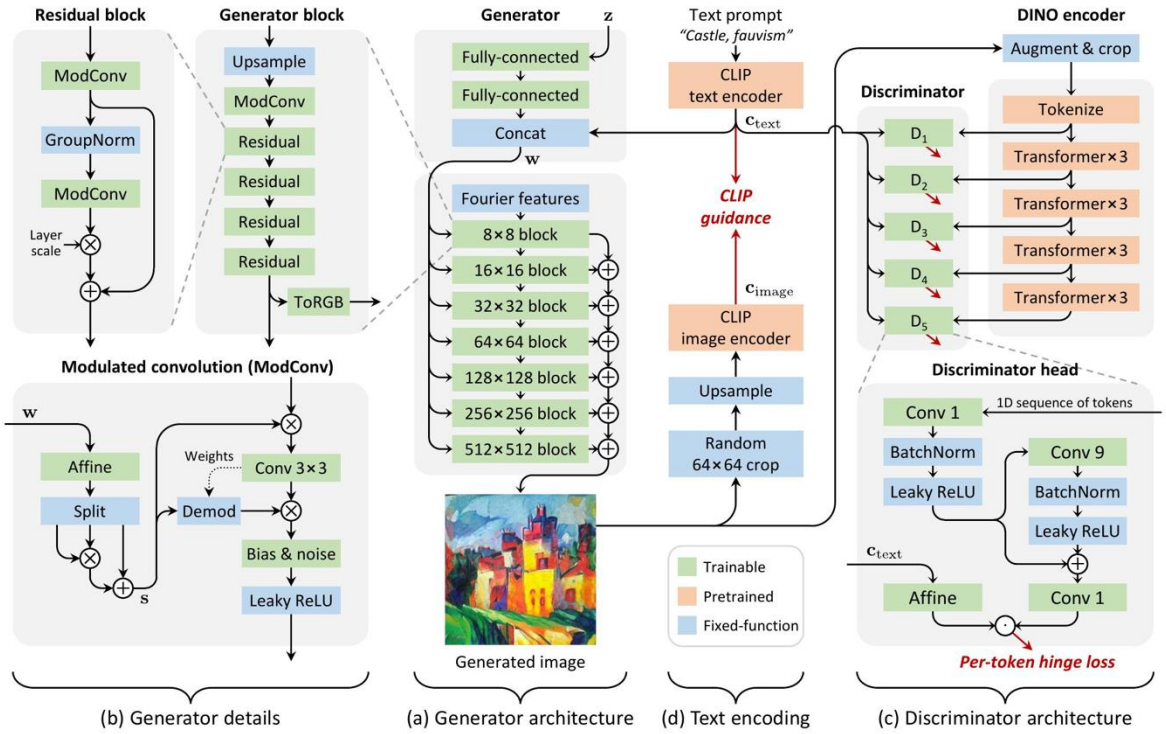


Figure 3: overview of StyleGAN-T

(a) Generator
StyleGAN2 backbone, blocks from 8×8 to 512×512, each adding to the image through its own ToRGB layer.

(b) Generator details
Residual blocks with GroupNorm and Layer Scale, and the ModConv unit that turns w into per-layer styles.

(c) Discriminator
A frozen DINO ViT-S trunk with five identical heads; text conditioning by projection at the very end.

(d) Text encoding
CLIP embeds the prompt, feeding the generator, the discriminator, and the guidance loss.

Generator: Macro Architecture

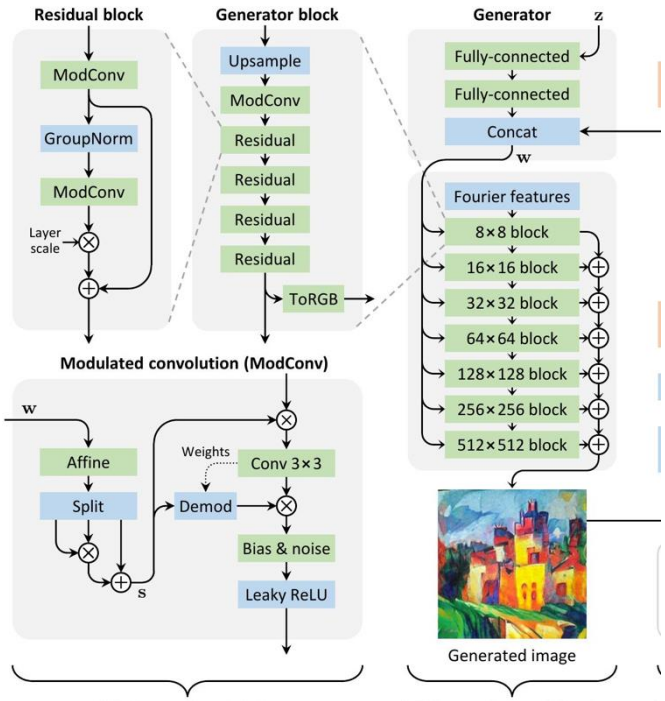


Figure 3 (a) and (b)

Equivariance is dropped, back to a StyleGAN2 backbone
Equivariance means features have no preferred pixel position.

Fourier features instead of a learned constant

Output skip connections: every resolution writes pixels

Generator: Residual Convolutions

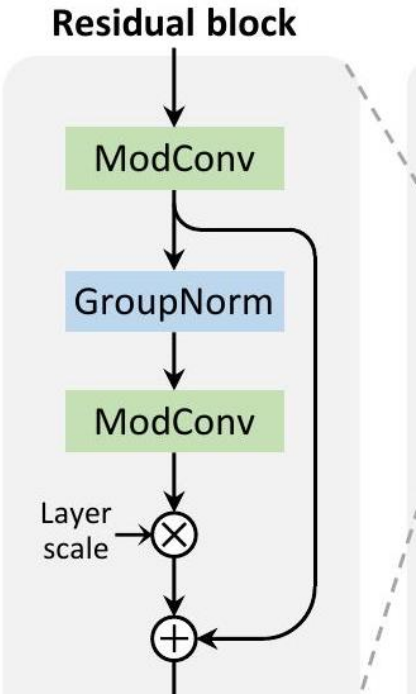


Figure 3 (b), residual block

The problem: depth caused early mode collapse

Absorbing 250M pairs needs capacity in both width and depth, but simply deepening a StyleGAN2 generator collapsed in the first iterations.

GroupNorm normalizes, Layer Scale scales

GroupNorm standardizes activations over channel groups; it is chosen over BatchNorm because it needs no communication across the 64 GPUs. Layer Scale is a learned per-channel multiplier on the branch output, from CaiT.

Layer Scale initialized at 10^{-5} is the crux

At initialization every residual branch contributes almost nothing, so the deep network behaves like the shallow, known-stable one; depth then fades in as the scales grow. This is what allows 2.3× more layers in the lightweight config and 4.5× in the final model, at matched parameter count.

Generator: Stronger Conditioning

The problem: z was drowning out the text

Fix 1: text bypasses the mapping network

Fix 2: second-order styles

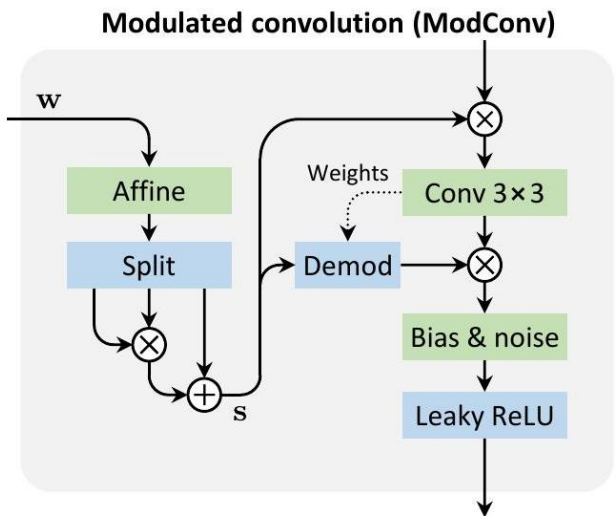


Figure 3 (b), modulated convolution

$$s = \tilde{s}_1 \odot \tilde{s}_2 + \tilde{s}_3 \qquad \text{(Eq. 1)}$$

Text Encoding: The Hub of the Diagram

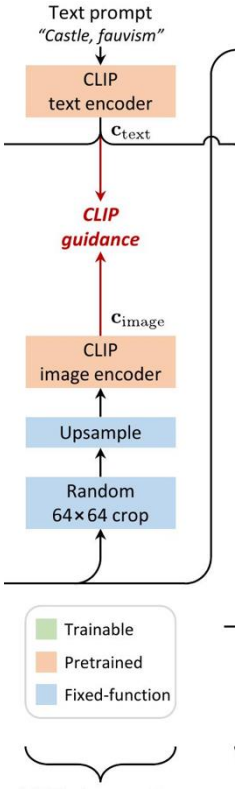


Figure 3 (d)

c_text fans out to three consumers

Into the generator's Concat box (the conditioning that shapes the image); into the discriminator heads (projection conditioning); and into the guidance loss, where it is compared against the image embedding. z has only one path — that asymmetry is the conditioning argument rendered as topology.

The guidance branch runs bottom-up

Generated image → random 64×64 crop → upsample → CLIP image encoder → c_image. The crop exists because guidance only helps up to 64×64; the upsample exists because CLIP ViT-L/14 expects 224×224 input. So the guidance signal is always computed on a small upsampled patch, never the full image.

Three different CLIP roles — keep them separate

Conditioning uses CLIP ViT-L/14's text tower (frozen in the primary phase). The guidance loss uses the original frozen CLIP image and text encoders. Evaluation uses a completely different model, ViT-g-14 trained on LAION-2B, so the reported CLIP score is not inflated by optimizing against the metric.

Discriminator: Architecture

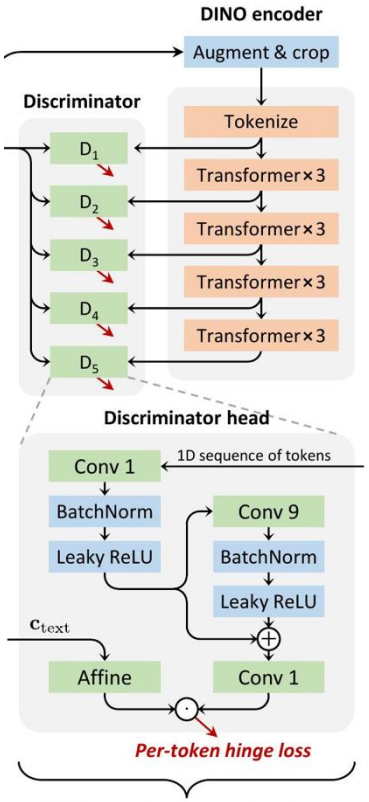


Figure 3 (c)

Feature network: a frozen DINO ViT-S/16

DINO is a self-supervised objective (self-distillation, no labels), and its patch tokens are strong dense descriptors: semantic information at high spatial resolution. It is also small and fast. y.

Isotropy is what lets the design be this simple

Five heads, spaced every three transformer layers

ViT-S has 12 layers, so heads sit at depths 0, 3, 6, 9, 12. D_1 reads the raw patch embeddings before any transformer runs. Close to local pixel statistics, while D_5 judges fully contextualized global semantics.

Discriminator: Inside One Head

All convolutions are 1D over the token sequence

The ⊕ before the loss is a projection discriminator

Per-token hinge loss, and BatchNorm that isn't BatchNorm

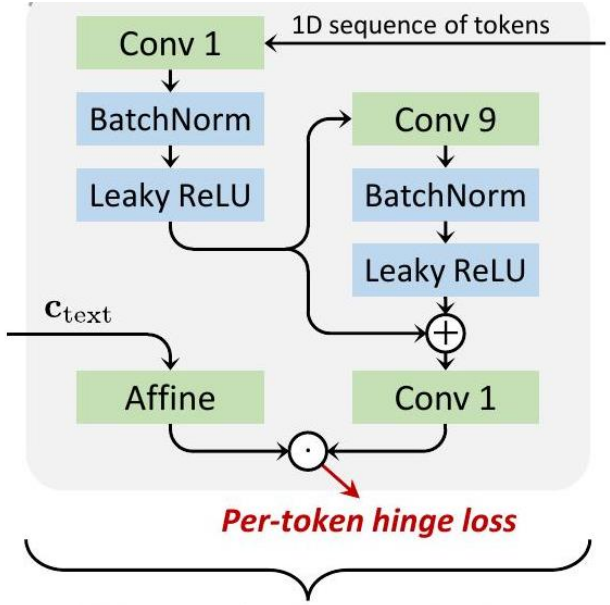


Figure 3 (c), discriminator head

CLIP Guidance: A GAN Analogue of Guidance

$$\mathcal{L}_{CLIP} = \arccos^2(c_{image} \cdot c_{text}) \qquad \text{(Eq. 2)}$$

Why guidance at all?

The novelty: guidance inside the training loss

What $\arccos^2$ measures

Guiding the Text Encoder: A Second Phase

	Primary phase	Secondary phase
Generator	trainable	**frozen**
CLIP text encoder	frozen	**trainable**
CLIP guidance weight	0.2	**50**
Duration	3 weeks + 5 days	2 days

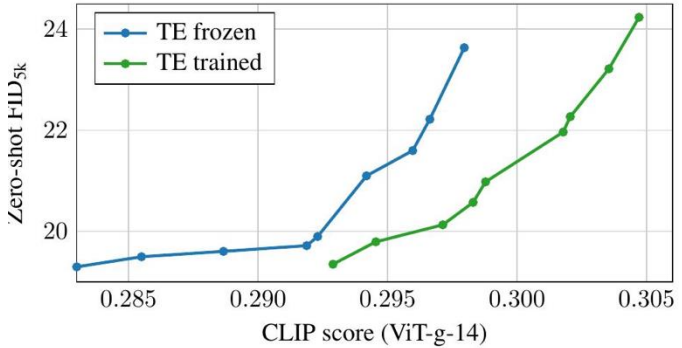


Figure 6: training the text encoder shifts the whole curve right

Result: alignment up, FID unchanged
The whole FID–CLIP curve moves right, so the CLIP score improves at every FID level. Because the generator was frozen, it handles both the original and fine-tuned embeddings equally well by FID, which the authors use as a sanity check that nothing was damaged.

Explicit Truncation: An Inference-Time Dial

The classic truncation trick
Sample w, then interpolate it toward the mean of W. The mean lies in a high-density region where the generator is well trained, so fidelity rises and diversity falls. ψ = 1 is untruncated; ψ → 0 goes to the mean.

Here the mean is per-prompt, not global

Measured effect: alignment up, variation down
For the Figure 4 prompt, mean CLIP score goes 0.33 → 0.36 → 0.39 as ψ drops from 1.00 to 0.10. At ψ = 0.10 the four samples per row are visibly near-identical, that is the diversity you paid.

Figure 4: "a graphite sketch of Eva Longoria" at ψ = 1.00 / 0.60 / 0.10

w̃ = 𝔼_z[w] = [f̃ , c_text] → interpolate w ↔ w̃ by ψ ∈ [0,1]

Training the Full Model

Phase	Resolution	A100-days	Iterations
Primary (3 weeks)	16×16	450	118,000
	32×32	450	78,000
	64×64	450	57,000
Secondary (2 days)	**text encoder**	**190**	**20,000**
Primary (5 days)	128×128	96	10,000
	256×256	70	6,000
	512×512	30	3,000

Appendix Table 5. One iteration = 2048 real + 2048 generated examples.

Results: MS COCO 64×64

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. Inference speeds measured on an A100; LAFITE's speed is estimated at native 64×64.

Results: MS COCO 256×256

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. Imagen and Parti use a faster TPUv4, which flatters their speeds.

Halves the previous GAN result
LAFITE 26.94 → StyleGAN-T 13.90, and still by far the fastest model in the table at 0.10 s.

But it trails diffusion and ARMs
eDiff-I reaches 6.95, Imagen 7.27, Parti-20B 7.23. StyleGAN-T sits between LDM and Ernie-ViLG.

Variation vs. Text Alignment

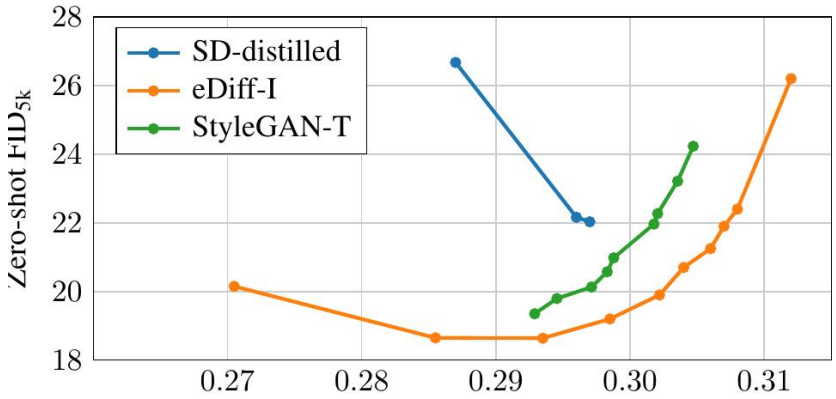


Figure 5: FID–CLIP curves, each model swept along its own knob

Beats the fast baseline on both axes

StyleGAN-T dominates SD-distilled, the previous state of the art in fast text-to-image in FID and CLIP score simultaneously , and pushes CLIP score to 0.305. It remains behind eDiff-I on quality.

And on speed it beats both

eDiff-I 32.0 s; SD-distilled 0.6 s at its best (8 steps); StyleGAN-T 0.1 s. So it is 6× faster than the model it outperforms, and 320× faster than the model that outperforms it.

Qualitative: Smooth Interpolation

Figure 2 (detail): one continuous interpolation read in scanline order. 56 such samples at 512×512 take 6 s on an RTX 3090.

Interpolating between prompts is a substitution, not a search: $w_0 = [f(z), c\_text0] \rightarrow w_1 = [f(z), c\_text1]$

Qualitative: Latent Control and Styles

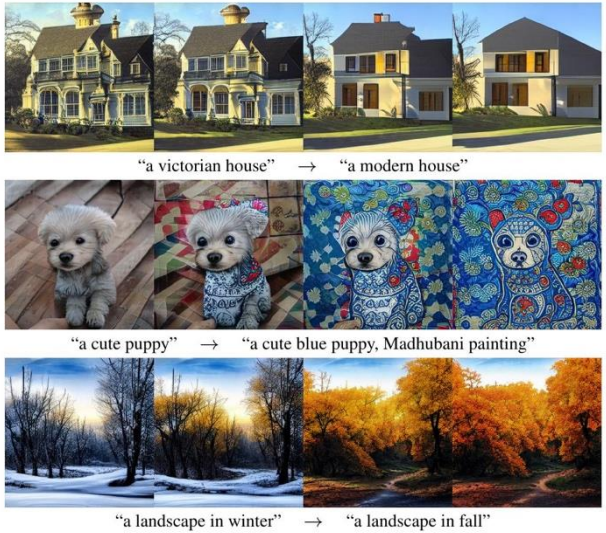


Figure 7: following semantic directions in latent space

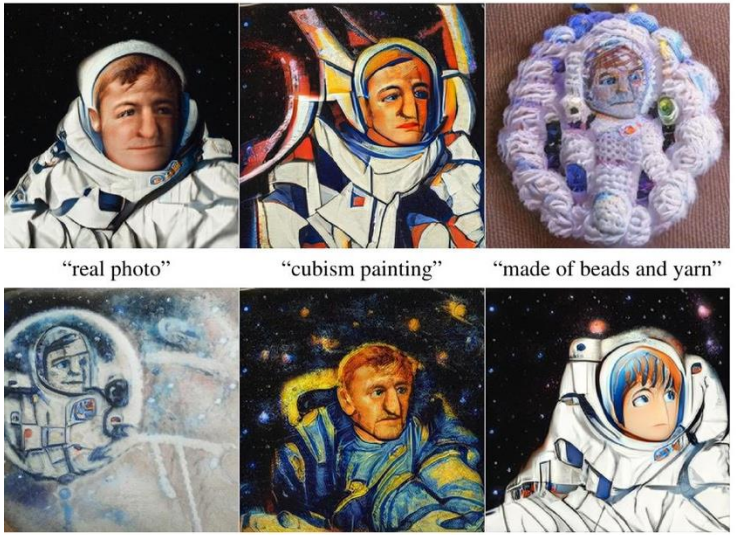


Figure 8: "astronaut, {X}" for one fixed random seed

Latent directions are free here

Styles disentangle from subject

Failure Cases

"a red cube on a blue cube" | "astronaut, child's drawing" | "a sign that says deep learning"

Figure 9: attribute binding and text rendering both fail

Attribute binding: "a red cube on a blue cube"
Colours and the spatial relation get scrambled. The model knows the objects and the colours but cannot reliably bind one to the other.

Coherent text in images: "a sign that says deep learning"
Letters come out mangled or mirrored. This is a known weakness of CLIP-conditioned models rather than of GANs specifically.

The root cause is the language model, not the generator
CLIP's text embedding is a single pooled vector optimized for retrieval, which compresses away compositional structure. DALL·E 2 shares this weakness for the same reason; Imagen and eDiff-I use T5-XXL and do better. A larger encoder would likely fix it, at the cost of runtime, which undercuts the GAN's one advantage.

Strengths and Weaknesses

Strengths

1. State of the art at 64×64 FID 7.30, while generating an image in 0.06 s, about 180× faster than the nearest diffusion model.

2. Beats distilled Stable Diffusion on FID and CLIP score simultaneously and is 6× faster than it.

3. Scales to ~1B parameters with no instability and no hyperparameter retuning, the historical GAN objection, answered empirically.

4. Retains what diffusion does not have: single-pass inference, genuinely smooth interpolation, and free semantic latent directions.

5. The ablation isolates where the difficulty lives, the discriminator, which is a transferable finding, not just a better number.

Weaknesses

1. Still behind diffusion at 256×256 (13.90 vs 6.95); the super-resolution stages are undertrained and the gap is unresolved.

2. CLIP as the language model means weak attribute binding and no coherent text rendering; fixing it costs the speed advantage.

3. CLIP guidance is essential but fragile, high strength introduces artifacts, so the weight must be hand-tuned to 0.2.

4. Truncation is not guidance: it is always toward a single mode, and it sharpens the distribution before the synthesis network, which can undo the sharpening.

5. Conclusions rest on FID and CLIP score, with no human evaluation of perceptual quality or prompt faithfulness.

Open Questions and Future Ideas

1. Truncation is not really guidance, can we do better?

Two structural differences the authors point out. Truncation is always toward a single mode, whereas diffusion guidance can in principle be arbitrarily multi-modal. And truncation sharpens the distribution in W, before the synthesis network, which can then reshape it arbitrarily and possibly undo the sharpening. A mechanism that acts on the output distribution directly could be strictly better.

2. A CLIP retrained on clean, high-resolution data

High guidance strength produces artifacts partly because CLIP was trained on aliased, low-quality web images, so optimizing against it rewards those defects. Retraining CLIP on higher-resolution data without aliasing might let the guidance weight go up safely. The authors also flag the discriminator's conditioning mechanism as worth revisiting.

3. A larger text encoder without losing the speed

T5-XXL would fix attribute binding but running it per prompt erodes the one-forward-pass advantage. Could the text embedding be distilled, or computed once and cached, so the language model cost is paid outside the generation loop?

4. Personalization, and the question this line of work actually answered

DreamBooth-style finetuning should transfer to GANs. More broadly: StyleGAN-T was largely superseded by adversarial distillation of diffusion models (ADD, SDXL-Turbo) from the same first author, which suggests the durable lesson was about the adversarial loss, not about GANs as a model family.

Student Questions

1. Why does StyleGAN-T do well at 64×64 but struggle at higher resolutions?

2. How do you balance text alignment against groundedness in real images?

3. Is it still meaningful to try to make GANs great again?

Key Takeaways

GANs were not fundamentally unsuited to large-scale text-to-image.
Three specific things were broken: the generator could not be deepened without collapsing, the discriminator's feature space was wrong for extremely diverse data, and there was no GAN analogue of guidance.

The discriminator was the dominant fix.
A frozen DINO ViT-S with five identical heads and a per-token hinge loss cut FID by 41% and raised CLIP score by 62%, more than everything else combined, and 2.5× faster than what it replaced.

Depth became trainable through Layer Scale at 10^{-5}.
Residual blocks that start as near-identity let the model grow 4.5× deeper and reach ~1B parameters with no instability and no retuning.

Text had to be given more and shorter paths to the output.
Bypassing the mapping network, second-order styles, and projection conditioning in the discriminator, all so that c_text stops being dominated by z.

THANK YOU