

Chitra Tiny: a 212M-parameter text-to-image model trained from scratch for about \$100

UDGAM LABS VERSION 1.0.0

Abstract

Chitra Tiny is a 212M-parameter English text-to-image model. Udgam Labs trained it from scratch and released it with open weights under Apache-2.0 on 12 October 2026. It is a latent diffusion transformer [1, 5] trained with rectified flow [2, 3]:

- A frozen, open text encoder (Flan-T5-Base) reads the prompt.
- A frozen, open image decoder (sd-vae-ft-mse) turns the result into a 256×256 picture.
- Only the transformer between them was trained. It started from random weights and learned from about 4.7 million public-domain and CCO images with automatically generated captions.

The whole project cost **\$100.46** of billed GPU compute, every experiment included. The final training run took about **14 hours on one H100 GPU**.

Results: - On 5,000 held-out images of the kind it was trained on, it reaches an FID of 22.1. - Its CLIP score is 26.2, close to the 27.5 that the real images score with their own captions. - On GenEval, an object-centric test of prompt following, it scores **0.219** at its default guidance of 7 (0.190 at guidance 4). That is about half of Stable Diffusion 1.5's 0.426 in our same harness, and below the 0.25–0.35 we had targeted before training.

It is good at photographs, paintings, landscapes, buildings, vehicles and food. It is weak at less common everyday objects, several objects in one picture, counting, positions, people and legible text. The 0.85 GB model makes an image in about **2.6 seconds** on the GPU of an Apple M6 Mac mini and in about **12 seconds** on its CPU, at the default 50 steps.

1. Introduction

Image generators are usually described as expensive to train. A widely cited market-rate estimate put the original Stable Diffusion's training at about \$600K [EXTERNAL; 20, 21]. Recent public reports show that much smaller models at 256 px can be trained from scratch for far less [EXTERNAL; Section 5.6]: - SupraLabs' Supra2-IMG, a 104M diffusion transformer, was trained on one H100 in about 9 hours, including data preparation [16]. - Logolabs' Agate Preview 001, a 260M-parameter model, reached GenEval 0.550 with 144.7 GH200-hours of training [15].

Both train on a public set of about 6 million synthetic images generated by FLUX.1 [dev] [15, 16, 18]. In our reading, the FLUX.1 [dev] licence does not allow its outputs to be used to train a model that competes with it [19].

We asked a narrower question: can a small team train a useful, general-purpose text-to-image model from scratch for about \$100, using only licence-clean images, and release it under a permissive licence? We fixed four constraints in advance:

- **Budget:** about \$100 of GPU compute for the whole project, not just the final run. Feasibility checks, pilot comparisons, data preparation, evaluation and publishing all count.
- **Data:** only images that their creators released into the public domain or under CC0, so the weights can be released under Apache-2.0.
- **Scope:** a general model (photographs, art, places, objects) rather than one domain, in English, at 256×256.
- **Use:** small enough to run on a laptop.

The short answer is **partly**: - **Where it works:** the kinds of pictures that fill public-domain collections (landscapes, architecture, paintings, vehicles, food). Chitra Tiny makes convincing, on-topic images of these, and they match their captions nearly as well as real photographs do. - **Where it falls short:** prompts that ask for specific everyday objects, several objects, counts or positions. Here it is far behind larger models, mostly because licence-clean collections contain few of those objects.

This report covers: - the model and its architecture (Sections 2–3); - training at a high level, with a cost ledger (Section 4); - evaluation, with every setting stated, plus speed and memory (Section 5); - how we present curated and uncurated samples (Section 6); - limitations and safety (Section 7); - what is released (Section 8).

2. Model overview

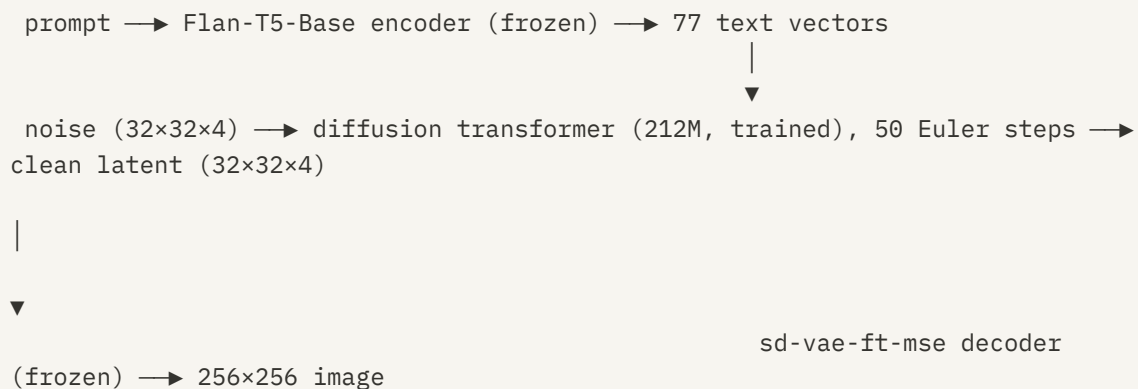
Developer	Udgam Labs
Released	12 October 2026, open weights
Task	English prompt → 256×256 RGB image
Trained parameters	212M (211.65M): a diffusion transformer, 16 layers, width 768
Pipeline parameters as loaded	404.93M: the transformer, plus the Flan-T5-Base encoder (109.63M), plus the full sd-vae-ft-mse autoencoder (83.65M). 370.77M of these are used to make an image; the autoencoder's 34.16M-parameter encoder half is loaded but unused
Method	latent rectified-flow diffusion with classifier-free guidance, trained with the Muon optimizer
Text encoder	google/flan-t5-base, frozen (Apache-2.0); not counted in the 212M
Image decoder	stabilityai/sd-vae-ft-mse, frozen (MIT); not counted in the 212M
Training data	about 4.7 million public-domain and CC0 images with automatically generated captions
Training compute	final run: 13 h 58 min on one H100; whole project: \$100.46 of GPU compute
Weights	0.85 GB, one fp32 safetensors file (exponential-moving-average weights)
Speed	about 2.6 s per image at 50 steps on an Apple M6 Mac mini's GPU, and about 12 s on its CPU, including text encoding and decoding (Section 5.7)
Licence	Apache-2.0

What "trained from scratch" means here. The diffusion transformer is the part that makes the picture. It started from random weights and learned only from our training images; no weights came from any other image model. It works with two existing open models, used unchanged: Flan-T5-Base reads the prompt, and sd-vae-ft-mse turns the result into pixels. The 212M figure counts only the trained transformer. With both helpers loaded by the released `chitra_tiny.py`, the pipeline has 404.93M parameters:

PART	PARAMETERS	TRAINED BY
Chitra Tiny diffusion transformer	211.65M	Udgam Labs, from scratch
Flan-T5-Base encoder (<code>T5EncoderModel</code> , encoder half only)	109.63M	Google, used frozen
sd-vae-ft-mse autoencoder, as loaded	83.65M (decoder 49.49M, used; encoder 34.16M, loaded but unused)	Stability AI, used frozen
Total as loaded	404.93M	

Chitra (चित्र) means "picture".

3. Architecture



1. **Text encoder.** The encoder half of Flan-T5-Base [6] turns the prompt into 77 vectors of width 768. Longer prompts are cut at 77 tokens; shorter ones are padded and masked.
2. **Latent space.** The frozen autoencoder [5, 7] represents a 256×256 image as a 32×32 grid with 4 channels (scaled by 0.18215). The transformer works only on this small grid, which keeps training and sampling cheap.
3. **Diffusion transformer** [1]. It cuts the 32×32×4 latent into 2×2 patches, giving 256 tokens of width 768 with fixed 2-D sine–cosine position embeddings. It then applies 16 blocks, each with 12 attention heads:
 4. self-attention over the image tokens;
 5. cross-attention from the image tokens to the text vectors;
 6. an MLP with 4× expansion.

Queries and keys are RMS-normalised in every attention layer. A conditioning vector, the timestep embedding plus the mean of the projected text vectors, sets a shift, scale and gate for

the self-attention and MLP sub-layers and for the output layer. The model predicts the velocity from image to noise. The exact layout is in the released `chitra_tiny.py` and `config.json`. 4. **Rectified flow** [2, 3]. Training places an image's latent and random noise at the two ends of a straight path, $x_t = (1 - t) \cdot x_0 + t \cdot \text{noise}$. The model learns the direction along that path. Sampling starts from pure noise ($t = 1$) and follows the predicted direction to the image ($t = 0$), in 50 equal Euler steps. 5. **Classifier-free guidance** [4]. At every step the model predicts twice, in one batch: once with the prompt, and once with an empty prompt (or the user's negative prompt). The sampler then moves further in the prompt's direction, scaled by the guidance value. 6. **Image decoder**. `sd-vae-ft-mse` turns the final latent into a 256×256 RGB image.

Inference settings

SETTING	DEFAULT	EFFECT
Guidance	7	Higher follows the prompt more literally; 3–5 gives more variety
Steps	50	25 is about 1.9× faster (Section 5.7), with a small loss of detail
Seed	0	Changes the picture; same seed and settings give the same picture on the same device
Negative prompt	empty	Things to steer away from; replaces the empty prompt in guidance

Why this shape: - **Borrowed helpers.** Using a frozen open text encoder and decoder means the budget goes only to the part that learns to paint. - **A small canvas.** A 256×256 picture, compressed to a 32×32 grid, keeps each training example small. - **Encode once.** The autoencoder is frozen, so we encoded every training image to its latent once, before training. The GPU then spent the main run on learning, not on processing images.

4. Training overview

4.1 Stages

STAGE	WHAT HAPPENED
1. Feasibility checks	We checked the licence of every component and tested data access and training throughput. We validated the GenEval harness by reproducing Stable Diffusion 1.5's published score (Section 5.3), and set a GenEval target of 0.25–0.35 for the final model
2. Data preparation	About 4.7 million images were resized and centre-cropped to 256×256 and encoded once into latents. Held-out images were set aside before training. We checked how often GenEval's object categories appear in the captions
3. Recipe selection	Short pilot runs ("bake-offs") of a few candidate recipes, at small model size on a subset of the data. Each candidate got the same compute, and the rules for choosing a winner were written down before any results came in. The project owner then signed off the recipe and the budget for the main run
4. Final run	One training run of the 212M model on one H100 GPU: 13 h 58 min of training on 11 October 2026
5. Evaluation	FID, CLIP and GenEval on the final weights (Section 5)
6. Release review	Before anything went public, the owner approved the exact results, the model card wording, a licence review and the release surfaces

Data. The training images come from curated public-domain and CCO collections. The collections' curators supplied automatically generated captions, which we did not write. These collections over-represent historic, Western and museum imagery. Before training we counted how often GenEval's 80 everyday object categories appear in the captions: many are rare. That count predicted the GenEval weakness we report in Section 5.4. The training data is not redistributed.

Recipe. The selected recipe trains the transformer with rectified flow [2, 3] and the Muon optimizer [8]. We do not publish the recipe's other hyperparameters.

Final run. During the run, we made a fixed grid of 16 prompts from saved checkpoints to track progress: - At 20% of the run, 10 of the 16 prompts already gave recognisable, on-topic images. - Animals and people were still rough at that point. - "A woman holding an umbrella in the rain" worked only at the end of training.

The released file holds the exponential moving average (EMA) of the weights.

4.2 Compute and cost ledger

ITEM	BILLED GPU COMPUTE
Whole project: every rented GPU machine on Runpod (42 machines)	\$100.46 (exact, from Runpod's billing records)
of which: recipe-selection bake-offs	~\$22
of which: the final training run (about 14 h on one H100)	~\$56
of which: everything else (feasibility checks, data preparation, rehearsals, evaluation, publishing, setup and idle time)	~\$23

Notes: - **The total is exact.** It comes from Runpod's billing records and includes every machine, including machines that failed to start and idle time. - **The split is approximate.** It comes from our per-machine log, which does not reconcile exactly with the bill. The rounded parts add up to about \$101. - **Not included.** Storage subscriptions, launch-media tools and online-demo hosting are not GPU compute and are not in the total. - **Context [EXTERNAL].** \$100.46 is about 1/6,000th of the ~\$600K estimate for the original Stable Diffusion [21], at about half of Stable Diffusion 1.5's GenEval score. That compares dollars across four years of GPU price changes; it is not a like-for-like compute comparison.

5. Evaluation

5.1 Setup

- **Weights and sampler.** Every number was measured by us on the released EMA weights, with the released sampler: 50 Euler steps, classifier-free guidance against the empty prompt.
- **Stable Diffusion 1.5** was run by us in the same GenEval harness.
- **Speed and memory** were measured by us on 17 October 2026 on an Apple M6 Mac mini (Section 5.7).
- **External numbers** are labelled [EXTERNAL] and cited; we did not reproduce them unless stated.

5.2 FID and CLIP score

METRIC	CHITRA TINY	REFERENCE	SETTINGS
FID [11], 5,000 held-out images (lower is better)	22.1	not comparable with published FID	guidance 4, 50 steps; one image per held-out caption; Inception features as in clean-fid [12], 256×256
CLIP score, ViT-L/14 (higher is better)	26.2	real held-out images with their own captions: 27.5	100 × image–caption cosine similarity, OpenAI CLIP ViT-L/14 weights [13, 14], same 5,000 captions

Notes: - **In-distribution.** The held-out images come from the same kind of collections as the training data and were set aside before training. FID here measures how well the model reproduces that kind of image. It is **not comparable** with FID on COCO or ImageNet, and it favours Chitra Tiny. - **CLIP.** A score of 26.2 against 27.5 for real images means the generated images match their captions nearly as well as the real photos do. - **Guidance.** FID was measured at guidance 4 only; we did not measure it at the default guidance of 7.

5.3 GenEval

We ran the official GenEval code and prompts [9]: - 553 prompts × 4 images (seeds 0–3), 50 steps. - Images are 256×256, upsampled to 512 for the Mask2Former object detector [10]. - Each task score is the fraction of images the detector judges correct. The overall score is the mean of the six task scores.

We validated the harness first. Stable Diffusion 1.5, run with the official generation script (50 steps, guidance 9), scored **0.426**, against **0.43** published by the GenEval authors.

TASK	CHITRA TINY, GUIDANCE 7 (DEFAULT)	CHITRA TINY, GUIDANCE 4	SD 1.5, OUR HARNESS	SD 1.5, PUBLISHED [EXTERNAL; 9]
Single object	0.60	0.56	0.96	0.97
Two objects	0.10	0.07	0.37	0.38
Counting	0.15	0.10	0.38	0.35
Colours	0.39	0.35	0.77	0.76
Position	0.01	0.02	0.04	0.04
Colour attribution	0.07	0.05	0.05	0.06
Overall	0.219	0.190	0.426	0.43

Against our target. During the feasibility checks, before training, we set a GenEval target of 0.25–0.35 for the final model. Chitra Tiny came in below it: 0.219 at guidance 7 and 0.190 at guidance 4.

Stable Diffusion 1.5's image model has about 860M parameters, about four times Chitra Tiny, and was trained on far more images [EXTERNAL; 20]. Both models are near zero on position and colour attribution, so the small differences there are not meaningful.

5.4 Why the gap: object coverage in the training data

GenEval asks for the 80 everyday object categories of COCO. We split those categories into thirds by how often each appears in our training captions, and measured single-object accuracy at guidance 4:

OBJECT FREQUENCY IN TRAINING CAPTIONS	SINGLE-OBJECT ACCURACY
Most common third	0.85
Middle third	0.51
Rarest third	0.33

Public-domain collections have many landscapes, buildings, paintings and vehicles, and few frisbees, toasters or hair driers. GenEval's short prompts ("a photo of a ...") also differ from the long, descriptive captions the model was trained on. We conclude that data coverage and model size, more than the sampler, set the GenEval score.

5.5 Sensitivity to guidance and prompt style

- **Guidance.** Raising guidance from 4 to 7 helped on every task except position (overall 0.190 → 0.219), so 7 is the release default.
- **Caveat.** We chose guidance 7 using the GenEval prompts themselves, so we always report 0.219 together with the guidance-4 result of 0.190.
- **Prompt style.** Rewriting GenEval's short prompts into the long descriptive style of the training captions *lowered* the score at both guidance values: 0.168 at guidance 4 and 0.177 at guidance 7. All headline numbers use the prompts exactly as written.

5.6 External context [EXTERNAL]

These numbers come from other teams' model cards, code and posts [15–17, 20, 21]. We did not run these models in our harness. Their training data, and in some rows the number of images per GenEval prompt, differ from ours.

MODEL	SIZE, RESOLUTION	REPORTED TRAINING COMPUTE	GENEVAL
Chitra Tiny (ours, measured)	212M, 256 px, latent	~14 H100-hours (final run); \$100.46 whole project	0.219 (guidance 7) / 0.190 (guidance 4)
Stable Diffusion 1.5 [20, 21]	~860M U-Net, 512 px, latent	150,000 A100-hours (SD v1 model card); ~\$600K at 2022 market rates	0.43 published [9]; 0.426 in our harness
Agate Preview 001 (Logolabs) [15]	260M (308.5M with the SD VAE decoder), 256 px, latent	144.7 GH200-hours	0.550
104M Supra2-style diffusion transformer (ablation in the Agate card) [15]	104M, 256 px, latent	13.1 GPU-hours	0.264 (1 image per prompt)
Supra2-IMG (SupraLabs) [16]	104.1M, 256 px, latent	~9 H100-hours, including data preparation	0.244 (1 image per prompt; measured by Logolabs [15])
MiniT2I-B/16 [17]	258M (+341M text encoder), 512 px, pixel space	not stated for this model (the smaller B/32 ablation model: about 3 days on 8 H100s)	0.873

How to read this table: - **Data matters at equal compute.** The Agate ablation used about as many GPU-hours as our final run and scored higher (0.264 vs 0.219). Supra2-IMG used fewer (about 9 H100-hours) and scored 0.244. Both were scored with one image per prompt (Logolabs measured SD 1.5 at 0.398 that way, against 0.43 with four [15]). Both were trained on synthetic images generated by FLUX.1 [dev] [18], which can cover GenEval's objects better than public-domain collections. - **More compute, other designs and other data go much further.** At similar sizes, Agate Preview 001 (about ten times our final run's GPU-hours) reaches 0.550. MiniT2I-B/16, a pixel-space model at 512 px trained on recaptioned web images, reaches 0.873. - **Chitra Tiny's place.** It sits at the low-cost, licence-clean end of this range and has the lowest GenEval score in the table. It is not at the quality frontier for its size. - **Not comparable.** Supra2-IMG reports FID 27.9 without stating its protocol, so we do not compare it with our FID.

5.7 Speed, memory, and what we did not measure

Speed and memory. We measured these on 17 October 2026 on a Mac mini with an Apple M6 chip (12-core CPU, 24 GB unified memory), with PyTorch 2.14.1 and the released fp32 weights and code. Each figure is the median of 7 images after one warm-up, at batch size 1 and guidance 7. Each time covers the whole call: text encoding, all sampling steps with guidance, and decoding to an image.

DEVICE	50 STEPS (DEFAULT)	25 STEPS	PEAK MEMORY
Apple M6 GPU (MPS)	2.6 s	1.4 s	3.3 GB peak memory footprint (Metal memory held: 2.8 GB; peak RSS 2.0 GB, reached while loading)
Apple M6 CPU (PyTorch's default threads)	12 s (11.8 and 12.3 s in two runs)	7–10 s (6.9 and 9.5 s in two runs)	3.0–3.2 GB peak RSS
NVIDIA GPU	not measured	not measured	not measured

Notes: - **A shared machine.** Other jobs were running on the Mac during these measurements. GPU times were stable to about 1%. CPU times varied by up to about 20% between images and runs; the slower 25-step run coincided with heavier background load. - **CPU threads.** On this chip PyTorch uses 2 threads by default. A run with 6 threads was not consistently faster under the same conditions (13.5 s at 50 steps, 5.7 s at 25 steps). Other CPUs, especially older or low-power laptop CPUs, may be considerably slower. - **Memory on Apple silicon.** With MPS the weights live in GPU (Metal) memory, which resident-set size does not fully count. We therefore report the kernel's peak memory footprint, which counts both.

Not measured: - speed and memory on NVIDIA GPUs; - human preference; - FID at guidance 7; - other small models in our own harness; - error rates of the demos' safety filters (Section 7.2).

6. Qualitative results policy

We show model outputs in two separate ways and label which is which. In both, the prompts were chosen by us. "Uncurated" means we did not pick the best of several images for a prompt.

Uncurated: what to expect

- **Walkthrough page** (udgamlabs.github.io/chitra-tiny):
- 12 gallery prompts, each run once with seed 0 (guidance 7, 50 steps).
- Five "things it gets wrong" prompts, also first try with seed 0: counting, text, position, faces and a rare object.
- A guidance dial (1, 3, 5, 7, 10) and a seed switcher (seeds 0–3) on fixed prompts.
- All 26 images come from a public script (`examples/make_walkthrough_images.py`), so anyone can regenerate them.
- Failures are captioned as failures. For example, "a bowl of soup next to a spoon" is marked "no spoon"; "three red apples on a table" shows two apples.
- **Model card grid:**

- The 16 fixed prompts we used to track training (Section 4.1), one image each: prompt i with seed i (seeds 0–15), guidance 7, 50 steps.
- Each prompt was generated once. Re-running the released code with these seeds reproduces the published grid exactly, pixel for pixel.
- **Live demos:** they show whatever the model makes for the visitor's prompt.

Selected outputs: launch material

The launch film, social posts and showcase grids use selected outputs: - 30 prompts were each generated with 8 seeds (0–7), at guidance 7 and 50 steps. - For each prompt we picked the best of the 8 images. 24 prompts were used. - Six prompts failed in all 8 seeds and were left out of launch material: "the Taj Mahal at sunrise", "a peacock in a garden", "an elephant walking through a forest", "a yellow auto rickshaw on a city street", "an astronaut floating above the earth" and "a japanese woodblock print of a great wave". - This material is labelled "selected outputs".

Rules for both

- Every image is a real Chitra Tiny output at 256×256.
- No other image or video model, and no stock footage, was used.
- The denoising sequences in the launch film are the model's real step-by-step predictions.

Judge typical quality from the uncurated set: best-of-8 selection overstates it.

7. Limitations and safety

7.1 Limitations

- **Less common everyday objects** (for example frisbees, toasters, carrots): 0.33 single-object accuracy on the rarest third of GenEval's objects.
- **Composition:** several objects in one picture, counting, and positions ("a cup left of a laptop").
- **People**, especially faces and hands.
- **Text:** it cannot write legible text in images.
- **Specific subjects:** in our launch selection, some named landmarks, animals and styles failed in all 8 seeds (Section 6).
- **Resolution and language:** 256×256 only, so fine detail is limited. English prompts only, cut at 77 tokens.
- **Bias:** training images and captions reflect their sources, which over-represent historic, Western and museum imagery. Expect these biases in outputs.

- **Scope of "licence-clean":** our transformer was trained only on public-domain and CCO images. The two frozen helper models were trained by their developers on other data, including web data. "Licence-clean" therefore describes our training images, not every component, and we do not call Chitra Tiny a "copyright-free" model.

Intended use: research, education, experiments with small diffusion models, and a starting point for fine-tuning.

Not suitable for: depicting real people, news or factual imagery, or any use where errors or biases could cause harm.

7.2 Safety

The model has no safety filter of its own. Its training images come from curated public-domain and CCO collections, but anyone deploying it publicly should add their own content moderation.

Both online demos (the Hugging Face Space and the try box on udgamlabs.com) add these measures:

- **Prompt blacklist.** Prompts and negative prompts are checked against a blacklist covering sexual content, graphic violence and self-harm, any mention of children or minors, and extremist terms. A blocked prompt gets a refusal message, and nothing is generated.
- **Output classifier.** Every generated image passes an open image-safety classifier, Falconsai/nsfw_image_detection [22], as stated in the Space's README. An image classified "nsfw" with a score above 0.5 is replaced by a "filtered by the safety check" tile, and the page says how many images were hidden.
- **Length caps.** Prompts are capped at 300 characters and negative prompts at 200 characters.
- **Rate limits (website try box).** Each visitor gets 15 generations an hour and 60 a day, keyed on a hashed visitor identifier; raw IP addresses are not stored.
- **Default negative prompt (website try box only).** When the visitor gives no negative prompt, the try box adds a fixed quality-oriented one. Its outputs can therefore differ from a local run with an empty negative prompt.

These measures are blunt: - The blacklist refuses harmless prompts that mention children, and paraphrases can get past it. - Classifiers miss things. - We have not measured how often either filter is wrong, in either direction.

8. Release details

SURFACE	CONTENTS
Hugging Face model udgamlabs/Chitra-Tiny	<code>model.safetensors</code> (EMA weights, fp32, 211.65M parameters, 846,620,520 bytes), <code>config.json</code> , <code>chitra_tiny.py</code> (standalone model and Euler/guidance sampler in PyTorch), <code>app.py</code> (local web playground), model card, sample grid, <code>LICENSE</code> . The text encoder and decoder are downloaded from their own repositories at run time, not bundled
GitHub udgamlabs/chitra-tiny (v1.0.0)	The same inference code (byte-identical), three runnable examples, <code>docs/EVALUATION.md</code> , <code>SHA256SUMS</code> , and the walkthrough page
Online demos	Hugging Face Space huggingface.co/spaces/jiteshkprasad/Chitra-Tiny ; no-login try box at udgamlabs.com/models/chitra-tiny
Udgam Labs website	udgamlabs.com
Not released	Training data, training code, intermediate checkpoints, optimizer state

Checks before release: - The release inference code and our training implementation give pixel-identical images for the same seed on CPU (maximum difference 0), with a strict weight load in which every key matched. - Every file's SHA-256 was matched against our manifest after upload. - From an empty cache with no login, the public weights downloaded and generated all 26 walkthrough images.

Reproducibility: noise comes from a CPU random generator with the given seed, so every device starts from the same noise. Results on different devices are near-identical, but not bit-identical.

```
pip install torch transformers diffusers safetensors huggingface_hub pillow
git clone https://huggingface.co/udgamlabs/Chitra-Tiny && cd Chitra-Tiny
python chitra_tiny.py "a lighthouse on a rocky coast at sunset" --n 4
```

Licence: - Weights and code: Apache-2.0. - Dependencies: google/flan-t5-base (Apache-2.0) and stabilityai/sd-vae-ft-mse (MIT). - Training data: the images were released by their creators into the public domain or under CC0.

```
@misc{chitratiny2026,
  title = {Chitra Tiny: a 212M text-to-image model trained from scratch},
  author = {{Udgam Labs}},
  year = {2026},
  url = {https://huggingface.co/udgamlabs/Chitra-Tiny}
}
```

Contact: Udgam Labs, via the Community tab of the Hugging Face model page or udgamlabs.com.

9. Conclusion

For \$100.46 of GPU compute, every experiment included, and a final run of about 14 hours on one H100, we trained a 212M text-to-image model from scratch on licence-clean images. It is released under Apache-2.0 and runs on a laptop: about 2.6 seconds per image on an Apple M6 Mac mini's GPU, and about 12 seconds on its CPU.

- **The budget is enough for a narrow kind of image.** On the scenes its training collections are full of, it makes convincing pictures that match their prompts nearly as well as real captioned photographs (CLIP 26.2 vs 27.5).
- **It is not enough for object-centric prompt following.** GenEval 0.219 missed our pre-training target of 0.25–0.35. It is about half of Stable Diffusion 1.5's score and well below recent models of similar size that used more compute or synthetic data.
- **Data coverage is the main lever.** Accuracy tracks how often an object appears in the training captions (0.85 for the most common third vs 0.33 for the rarest). The most useful next step is better licence-clean coverage of everyday objects, more than model tricks.

We publish the failures alongside the results and keep curated and uncurated samples clearly apart.

10. References

1. Peebles, W., Xie, S. *Scalable Diffusion Models with Transformers*. ICCV 2023. arXiv:2212.09748.
2. Liu, X., Gong, C., Liu, Q. *Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow*. ICLR 2023. arXiv:2209.03003.
3. Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., Le, M. *Flow Matching for Generative Modeling*. ICLR 2023. arXiv:2210.02747.
4. Ho, J., Salimans, T. *Classifier-Free Diffusion Guidance*. arXiv:2207.12598, 2022 (short version at the NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications).
5. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B. *High-Resolution Image Synthesis with Latent Diffusion Models*. CVPR 2022. arXiv:2112.10752.
6. Chung, H. W., Hou, L., Longpre, S., et al. *Scaling Instruction-Finetuned Language Models (Flan-T5)*. arXiv:2210.11416, 2022. Model: [google/flan-t5-base](https://huggingface.co/google/flan-t5-base).
7. Stability AI. *sd-vae-ft-mse* model card. huggingface.co/stabilityai/sd-vae-ft-mse.
8. Jordan, K., Jin, Y., Boza, V., You, J., Cesista, F., Newhouse, L., Bernstein, J. *Muon: An optimizer for hidden layers in neural networks*. Blog post, December 2024.

kellerjordan.github.io/posts/muon.

9. Ghosh, D., Hajishirzi, H., Schmidt, L. *GenEval: An Object-Focused Framework for Evaluating Text-to-Image Alignment*. NeurIPS 2023 Datasets and Benchmarks. arXiv:2310.11513. Code: github.com/djghosh13/geneval.
10. Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., Girdhar, R. *Masked-attention Mask Transformer for Universal Image Segmentation* (Mask2Former). CVPR 2022. arXiv:2112.01527.
11. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium* (FID). NeurIPS 2017. arXiv:1706.08500.
12. Parmar, G., Zhang, R., Zhu, J.-Y. *On Aliased Resizing and Surprising Subtleties in GAN Evaluation* (clean-fid). CVPR 2022. arXiv:2104.11222.
13. Radford, A., Kim, J. W., Hallacy, C., et al. *Learning Transferable Visual Models From Natural Language Supervision* (CLIP). ICML 2021. arXiv:2103.00020.
14. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y. *CLIPScore: A Reference-free Evaluation Metric for Image Captioning*. EMNLP 2021. arXiv:2104.08718.
15. Logolabs. *Agate Preview 001* model card, including its ablation and GenEval tables. Hugging Face, September 2026. huggingface.co/Logolabs/agate-preview-001 (accessed 17 October 2026).
16. SupraLabs. *Supra2-IMG* model card. Hugging Face, September 2026. huggingface.co/SupraLabs/Supra2-IMG (accessed 17 October 2026).
17. Wang, X., Zhao, H., Lu, Y., Zhou, K., Ma, L., He, K. *MiniT2I: A Minimalist Baseline for Text-to-Image Generation*. Blog post, 16 June 2026. peppaking8.github.io/minit2i. Code and GenEval table: github.com/Hope7Happiness/t2i-release (accessed 17 October 2026).
18. FLUX-Reason-6M dataset card (images generated by FLUX.1 [dev]). Hugging Face. huggingface.co/datasets/LucasFang/FLUX-Reason-6M (accessed 17 October 2026).
19. Black Forest Labs. *FLUX.1 [dev] Non-Commercial License*, section 2(d), "Outputs". github.com/black-forest-labs/flux/blob/main/model_licenses/LICENSE-FLUX1-dev (accessed 17 October 2026).
20. CompVis. *Stable Diffusion v1* model card, "Environmental Impact": A100 PCIe 40GB, 150,000 hours. huggingface.co/CompVis/stable-diffusion-v1-4; the same figures appear in the *Stable Diffusion v1.5* card. Model size (860M U-Net): github.com/CompVis/stable-diffusion.
21. Mostaque, E. (then CEO of Stability AI). Post on X, 28 August 2022, estimating Stable Diffusion v1's 150,000 A100-hours at about \$600K at market prices. x.com/EMostaque/status/1563870674111832066.

22. Falconsai. *nsfw_image_detection* model card. Hugging Face.

huggingface.co/Falconsai/nsfw_image_detection.

23. Udgam Labs. Chitra Tiny model card; GitHub repository, `docs/EVALUATION.md` and walkthrough page, October 2026.

About Udgam Labs. Udgam Labs is an independent AI research lab. We work from first principles: before we build, we understand why things work, whether we are training a model, fine-tuning one or applying AI to a new problem. Udgam (उद्गम) is Sanskrit for origin. udgamlabs.com