

Training a Text-to-Image Model from Scratch on Two GPUs

A Latent Diffusion Transformer on CC12M, a Small Real Prompt Effect, and the Bakeoffs That Moved the Loss but Not the Picture

Jovonni L. Pharr
Georgia Cyber Warfare Range / BlackMind
info@gacwr.org

September 2026

Abstract

We trained BlackMind’s first text-to-image model — a latent diffusion transformer — from random initialisation on two consumer GPUs, with no downloaded generator weights and no pretrained image prior. This is the M1 stage of a staged video-model research program: the image model is deliberately trained first, as the pipeline-validation rung on a falsifiable ladder, and its gate is explicit and modest — beat an SD1.5-class baseline on *our own* harness at matched inference compute, not approach the frontier. We report the result and, with equal weight, the measurement discipline that framed it. The port is correct: a DiT-S memorises 64 fixed images and routes text to the right one **64 of 64** times against a 1.6% chance baseline, with training loss falling $1.24 \rightarrow 0.0035$. The bakeoffs then do their job. A Muon-hybrid optimiser beats all-AdamW at equal wall-clock on held-out velocity-MSE (0.18333 versus 0.18609, a gap $9.2\times$ the seed-noise bound). The learning-rate sweep is nearly flat, and we keep the audit correction that retracted its first, over-stated verdict. But the sharpest result of the program is a measurement one: a CLIP arm-similarity study shows that three of six intervention pairs — including the *entire* learning-rate sweep — changed the picture no more than a change of random seed does; the interventions moved the loss but not the image. The optimiser verdict is perceptually under-powered, and the one intervention that demonstrably *did* move the picture is the one that *lost*. Classifier-free guidance was inadvertently left at $w=1.0$ for the entire run; a sweep shows that turning it on is a real, significant gain in prompt alignment but is mostly beautification, not comprehension, and we keep the correction that the shipped value was chosen on a non-significant screen. Finally, on 64 held-out prompts the model prefers its own caption’s ground truth 4 of 64 times against 1.6% chance ($p=0.018$) — a small but real conditioning effect. We keep every retraction on purpose: the discipline that kills a wrong claim is what makes a right one trustworthy.

Contents

1	Introduction	3
2	The model and the rig	3
3	The trainer, validated	5
4	The bakeoffs	6
4.1	Optimiser: Muon-hybrid beats all-AdamW at equal wall-clock	6
4.2	Learning rate: nearly flat, and the verdict we had to restate	6
5	The finding: the arms moved the loss, not the picture	7

6	Guidance	8
7	A small, real prompt effect	9
8	What we retracted, and the honest limits	9
9	What is unresolved	12

1 Introduction

This paper documents M1, BlackMind’s first image generator: a latent diffusion transformer (DiT) [1] trained from random initialisation on CC12M [8] images, on two consumer graphics cards. There are no downloaded generator weights, no LoRA, and no pretrained image prior at any point in the run. The only frozen components are a VAE and a text encoder, used strictly as fixed instruments; the transformer that does the drawing is ours, from scratch.

M1 is not a stand-alone project. It is the first real training rung of a staged video-model research program whose later rungs (M2 through M6) build 2-second, then 10-second, then audio-conditioned video on the same rig. The image model comes first by design, because the cheapest way to prove a diffusion-transformer training pipeline works is to make it draw a still frame before asking it to make a moving one. The program runs on a falsifiable ladder: every milestone is a pre-registered gate, passed only by an audited receipt, and failing a gate is treated as a result, not a delay. M1’s gate is written out in full and it is deliberately modest: *our $\sim 150\text{--}400M$ latent DiT, trained on our corpus, beats a frozen SD1.5-class open checkpoint on our suites under our judge protocol, at matched inference compute*. What the gate explicitly does *not* claim is parity with modern frontier image models. SD1.5 is a 2022-era 860M U-Net; beating it with a 2024-era DiT recipe and curated data is *expected if the pipeline works*. M1 is a pipeline-works gate. Failing it would mean the pipeline is broken, full stop.

Two things make this worth writing up. The first is the model itself. From random noise it learns to draw recognisable objects and scenes — cars, garments, food, rooms — with distortion and garbled text, but unmistakably on the right track (Figure 1). The second, and in our view just as valuable, is the measurement discipline we built around it. That discipline is sharp enough to produce the central finding of this paper: that most of the interventions we ran during the stage moved the training loss without moving the picture, that our most-cited optimiser result is perceptually under-powered, and that the one lever we can prove reached the pixels is the lever that made the metric worse. These are not comfortable findings. They are the point. We keep the retractions and audit corrections in the body on purpose — they are the proof the instruments work.

2 The model and the rig

Table 1 records the system exactly as configured. Every architectural and training choice named here is load-bearing for a claim later in the paper.

The generator is a latent diffusion transformer in the DiT family [1], operating on the latent space of a frozen autoencoder in the manner of latent diffusion [2]. The latents come from the Wan2.1 VAE [11]: 16-channel, 256×256 -image latents, posterior-mode and unscaled per the porting convention, with a `vae_hash` of `d6e524b3...` matching the program’s stage-0 VAE gate. The text pathway is Qwen3-1.7B [10], frozen in bf16, taking the `last_hidden_state` at up to 226 tokens, precomputed and cached so the trainer never runs the encoder at train time. The patchified sequence is 226 text tokens + 256 latent tokens (a 2×2 patchify of a $32 \times 32 \times 16$ latent), 482 tokens in all.

The objective is flow matching with a velocity-MSE loss [3], in the rectified-flow formulation [4] — a continuous-time successor to the discrete-time denoising diffusion formulation [13] — with logit-normal timestep sampling and the SD3-style train-time timestep shift [5]. This core — `src/train/flow_match.py`, with transport/path line receipts in comments — was ported from Tencent’s HunyuanVideo-I2V flow-matching implementation [12], and the sampler is a ported Euler scheme over the FlowMatchDiscrete scheduler semantics. The optimiser is a Muon-hybrid: Muon [9] drives the 2-D block weight matrices with Newton-Schulz orthogonalisation (momentum 0.95, 5 iterations), while AdamW drives everything else (adaLN, embeddings, patchify). The schedule is Warmup-Stable-Decay [14], with a sqrt cooldown over the final 20% of each

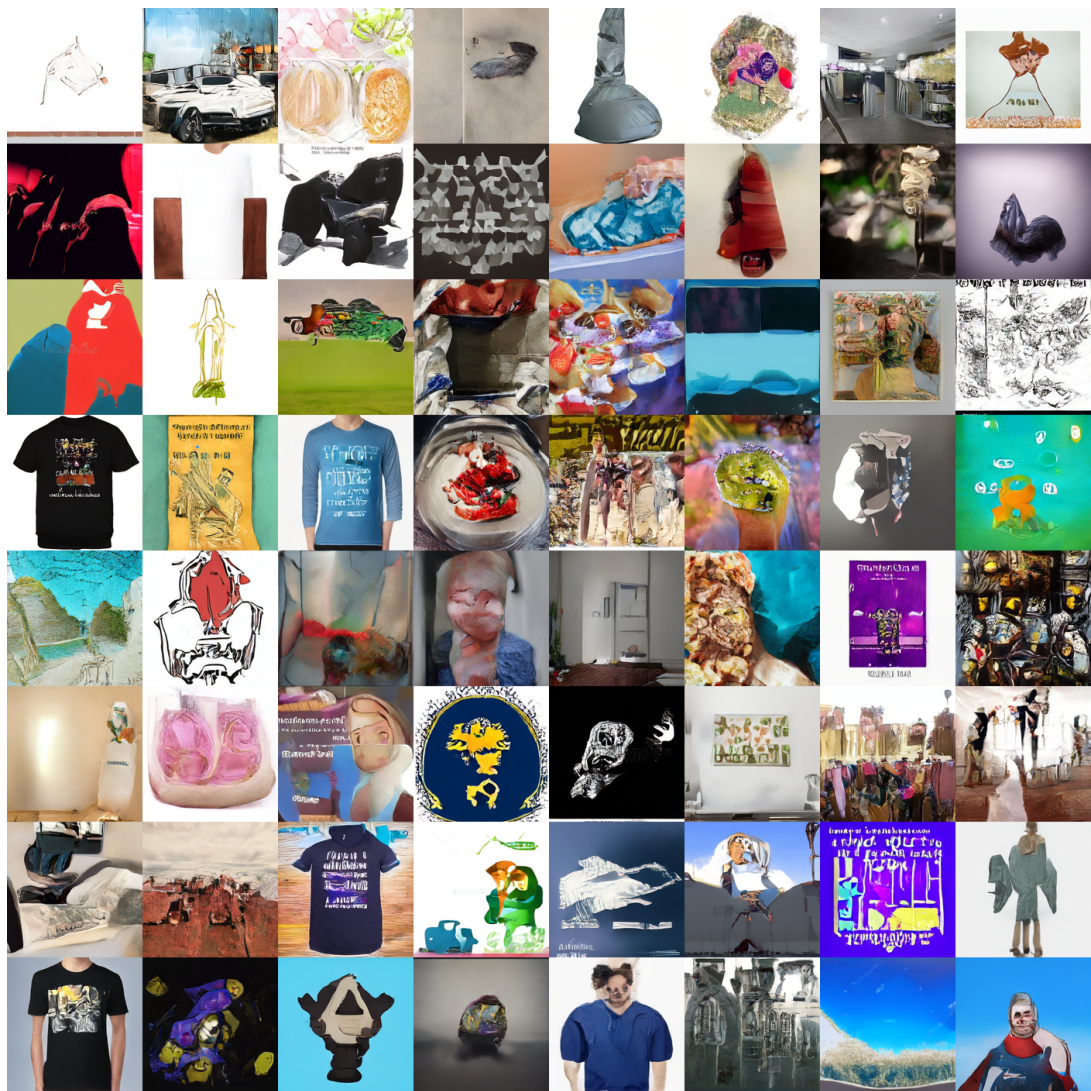


Figure 1: What the model draws. The 8×8 held-out generation grid from the Stage-1 checkpoint: 64 prompts the model never saw in training, sampled from pure noise. The output is recognisable — cars, garments, food, interiors — with structural distortion and garbled text. This is the honest state of the image model, shown before any number in the paper.

Table 1: The system under test. The generator is trained from random initialisation; the VAE and text encoder are frozen and used only as fixed instruments.

component	choice	note
generator	latent DiT (BAVDiT), DiT-S 43.67 M in the smoke; DiT-B 166.95 M for the stage	ours, from random init; presets S/B
objective	flow matching / rectified flow, velocity MSE, logit-normal t , SD3 train-time shift	core ported from HunyuanVideo-I2V
latents	frozen Wan2.1 VAE, 16-channel, 256×256	<code>vae_hash d6e524b3...</code> (G0 receipt)
text	frozen Qwen3-1.7B <code>last_hidden_state</code> (2048-d), ≤ 226 tokens	precomputed and cached; trainer never runs the encoder
optimizer	Muon on 2-D block matrices + AdamW on the rest, WSD schedule	warmup + sqrt cooldown over final 20%
data	CC12M, cached to latents; 97,952 train + 2,048 held-out	single-bucket batches, text padded to batch-max
scale	$2 \times$ RTX PRO 4000 Blackwell (24 GB)	one arm per GPU; grad-ckpt mb32 $\times$ 2

budget leg. All of this runs on two power-constrained RTX PRO 4000 Blackwell cards with 24 GB each.

A note on the parameter target. The program’s gate is written for a ~ 150 – 400 M model; the stage model is DiT-B at 166.95 M (the spec’s ~ 150 M estimate omitted the conditioning stack, which is ~ 20 M with a 2048-d text dimension and a $4 \times$ -MLP refiner — within the 150M class, trimmable to a $2 \times$ refiner if exact parity ever matters). The DiT-S preset used in the smoke is 43.67 M. Throughput receipts: DiT-S runs at 0.451 s/step (142 images/s) at 9.08 GB peak; DiT-B at 1.502 s/step (43 images/s) at 14.44 GB. DiT-B at micro-batch 32 OOMs (23.4 GB) without gradient checkpointing; with per-block activation checkpointing it fits at 4.20 GB (a +24% step-time cost), which is why every stage arm runs grad-checkpointed. The spec’s claim that “micro-batch 64 images fits at 150M” is refuted as written for this implementation until checkpointing or token-packing lands — a deviation we log rather than paper over.

3 The trainer, validated

Before any ranking claim, the trainer earns the right to be trusted by memorising a tiny fixed set and routing text correctly. This is a plumbing proof, and it passed cleanly.

Setup. 64 images from a CC12M shard (first 64 with non-empty captions), centre-cropped to 256×256 , encoded once through the frozen Wan2.1 VAE to latents [64, 16, 1, 32, 32], normalised per-channel with the Wan `latents_mean/std`. Captions encoded once through frozen Qwen3-1.7B. DiT-S (43.67 M). Objective: velocity MSE, logit-normal t , train-time shift 1.0, zero text-dropout (this is an overfit test). Muon lr 0.02 / AdamW-group lr $1.5e-3$, WSD with sqrt cooldown, grad clip 1.0, bf16 autocast.

Loss. Over 1000 steps at effective batch 64 (the full set every step), the 100-step smoothed loss falls from 1.2444 (mean of the first 20 steps) to 0.00353 at step 1000 — a $352 \times$ drop, monotone at every 100-step checkpoint, no spikes, no NaN, in 7.52 minutes of wall-clock. A dry-run receipt (20 steps $1.3358 \rightarrow 1.1865$, VAE-hash cross-check, a clean decoded thumbnail matching caption 1, 8.91 GB high-water) preceded it.

The go/no-go. Sampling 64 images from pure noise (seed 0, 50 Euler steps), each conditioned on one of the 64 cached caption embeddings and decoded through the VAE, every single sample is nearest in latent space to its *own* caption’s ground-truth image: **64 of 64**, 100%, **against a 1.6% chance baseline**. The decoded grid shows legible memorised text (“PLANS FOR 2019”, “GRAND PRIX”, “MERRY CHRISTMAS” posters) and correct scenes per cap-

tion. We ran the first-skeptic check: this is not a decode-the-GT bug, because samples start from pure noise and the mean latent L2 distance to the matching ground truth is 3.11, not ~ 0 — conditioning, not tensor identity, selects the image. The ported flow-matching core under the Muon+WSD harness is functionally correct. This smoke makes no claim about quality, generalisation, or LR optimality; those are the first real GPU-hours, below.

4 The bakeoffs

With the pipeline validated, the stage ran a set of two-arm bakeoffs, each with a pre-registered gate on held-out velocity-MSE (a fixed-noise, fixed-timestep, seed-4242 eval over the 2,048 held-out rows, comparable across steps). Two produced directional verdicts; one of them required an audit correction that we keep.

4.1 Optimiser: Muon-hybrid beats all-AdamW at equal wall-clock

Table 2: Optimiser bakeoff, 90-minute budget per arm, held-out velocity-MSE.

arm	optimiser	lr	final held-out MSE	steps
m1lr_1x (comparator)	Muon-hybrid	0.02 / $1.5e-3$	0.18333	3894
m1_bake_adamw_hi	all-AdamW	$1.5e-3$	0.18609	4232
m1_bake_adamw_lo	all-AdamW	$5e-4$	0.18663	4199

The pre-registered verdict fired: **Muon-hybrid wins at equal wall-clock**. The gap to the best AdamW arm is 0.00276, which is $9.2\times$ the seed-noise bound of 0.00030 measured elsewhere — far outside the “no advantage” clause. AdamW even completed $\sim 8\%$ more steps (no Newton-Schulz overhead: 0.78–0.80 it/s versus 0.73) and still lost, so its per-step disadvantage exceeds its throughput edge. Within AdamW, the hotter LR won weakly ($1.5e-3$ over $5e-4$ by 0.00054), unresolved. This matches the sibling LLM program’s Muon finding: the port’s hot-LR heritage transfers to flow-matching DiTs. The headline consequence is that Muon-hybrid at spec defaults is the Stage-1 recipe.

Audit correction (kept). The science audit downgraded three things about this result, and we keep them in the body because they bound the claim honestly. (1) The comparison is *not* a pure optimiser A/B: the Muon trunk matrices carried weight-decay 0.0 while all-AdamW carried 0.1, a regularisation confound on $\sim 85\%$ of the parameters — so this is “Muon-hybrid recipe vs an all-AdamW recipe,” not an isolated optimiser swap. (2) The “ $9.2\times$ seed noise” multiplier is withdrawn as a significance statement ($n=1$ per arm, $n=1$ -pair anchor); what stands directionally is that Muon led at every checkpoint, AdamW lost despite more steps, and both AdamW arms landed above all four Muon runs. (3) AdamW’s LR screen was two-point and downward-only; a $3e-3$ falsifier was never tried while the trend said hotter-is-better, so “recipe locked” is downgraded to “recipe provisionally selected pending the AdamW- $3e-3$ arm.”

4.2 Learning rate: nearly flat, and the verdict we had to restate

Table 3: Learning-rate sweep, 360-minute budget per arm, held-out velocity-MSE.

arm	muon-lr / adamw-lr	seed	final held-out MSE	steps
lr0.5x	0.01 / $7.5e-4$	1234	0.18291	3929
lr0.5x_seed2	0.01 / $7.5e-4$	5678	0.18261	3891
lr1x (default)	0.02 / $1.5e-3$	1234	0.18333	3894
lr2x	0.04 / $3.0e-3$	1234	0.18441	3926

The two seeds of the $0.5\times$ arm differ by $|0.18291 - 0.18261| = 0.00030$ — this is the seed-noise bound the pre-registration used as its yardstick. The $0.5\times$ versus $1\times$ gap is 0.00042, and the $2\times$ (0.04) arm is worst by 0.0011–0.0017 but never diverged: stable, past the optimum, consistently ordered last at every one of the 8 intermediate evals.

Audit correction (kept, and it is the point). The stage initially declared the LR “insensitive in this range” and let the spec default stand. The science audit found this *violated the pre-registered rule as written*: the rule required a gap $\leq$ the between-seed gap, and $0.00042 > 0.00030$ — the “insensitive” clause did not fire, and the “ $\sim 1.4\times$ ” softening was post-hoc. The restated verdict, under the letter of the pre-registration, is that **muon-lr 0.01 was the screening winner** (it beat 0.02 under both seeds); the earlier framing conveniently preserved the default and is retracted. The “ $3.6\text{--}5.6\times$ seed noise” multipliers on the $2\times$ deficit are also withdrawn ($n=1$ seed pair; the CI on the bound spans roughly $[0.45\times, 30\times]$, and the pair is confounded with GPU and a 38-step delta). The directional statements that survive are narrow and true: 0.04 is worst at all 8 eval points, and it is stable. We keep this correction because it is exactly the discipline the finding in the next section rests on.

5 The finding: the arms moved the loss, not the picture

Here is the result that, in the words of our own SOTA audit, “should stop the program”: every arm produces nearly the same picture. The audit asserted it from looking at grids. This section is the measurement.

Method. Every gallery grid is the same 16 held-out prompts at the same seed and sampler settings, so cell i of two grids is two models’ answer to one prompt. We encode each cell with CLIP ViT-L/14 [7] and take the mean cosine between matched cells, judged against two anchors measured identically: a *prompt floor* (different prompts, same model — how similar two images are when *everything* differs: 0.7330), and a *seed floor* (same recipe, different training seed — how similar two pictures are when *nothing* was varied). The two seed pairs give 0.7973 and 0.8503, mean 0.8238; they disagree by 0.0529, which is large next to the effects being judged, so a claim that an arm changed the picture must clear the *conservative* floor of ~ 0.85 , not the average.

Table 4: CLIP arm-similarity. Higher means more similar; the conservative seed floor is ~ 0.85 .

pair	similarity	CI95	verdict
lr0.5x vs lr1x	0.8738	[0.8608, 0.8891]	same as a seed change
adamw_hi vs adamw_lo	0.8611	[0.8400, 0.8825]	same as a seed change
lr0.5x vs lr2x	0.8612	[0.8289, 0.8888]	same as a seed change
adamw_hi vs lr1x (AdamW vs Muon)	0.8463	[0.8230, 0.8710]	ambiguous (CI straddles floor)
lr1x vs lr2x	0.8380	[0.7994, 0.8745]	ambiguous
inject_A vs inject_B	0.7835	[0.7538, 0.8142]	changed the picture

What this says. The audit’s claim is substantially confirmed, though not universally. Three of six arm pairs — *including the entire learning-rate sweep* — are as similar to each other as two runs of the same recipe with a different seed. Those interventions moved held-out velocity-MSE without moving the picture.

The sting is in the two headline results. First, our most-cited bakeoff outcome — Muon ahead at every checkpoint — is, perceptually, the *ambiguous* one: AdamW versus Muon scores 0.8463 with a CI straddling the conservative seed floor. It cannot be called a perceptual difference and it cannot be called a non-difference; it is under-powered. Second, the one intervention that demonstrably changed the picture is the one that *lost*: quality-injection (A vs B, 0.7835, below every seed pair) is the only arm pair that moved the image distribution, and its pre-registered

verdict was that injection *hurt* (0.18226 vs 0.17879 on the primary metric). The single lever that reached the pixels made the metric worse.

Limits, stated before anyone quotes this. $n=16$ prompts, one grid per arm; the CIs are bootstraps over those 16. The seed floor rests on 2 pairs that disagree by 0.053, comparable to the entire spread of arm effects — this measurement is under-powered by construction and no verdict here is settled. Arms are compared at latest checkpoints that differ slightly in step (3,891–4,232). CLIP cosine is not human judgment; it is a screening instrument, and it is only the second metric this program has ever had. The stronger version — 64-prompt grids and ≥ 4 seed pairs — is queued. The lesson is not that the loss metric is useless; it is that a loss delta is not a picture delta, and this program now knows to demand both.

6 Guidance

An adversarial audit found that the trainer spent 10% of every batch training an unconditional branch (text-dropout 0.1) that the sampler never used: **every image this program ever published was rendered at classifier-free guidance $w=1.0$** [6] — that is, with guidance effectively off. Fixing it costs zero GPU-hours of training; only the sampler changes. We ran a sweep on the same checkpoint, the same 16 held-out prompts, the same seed, 50 Euler steps.

The right criterion is not saturation but *does the picture depend on its prompt*, measured by a paired matched-minus-mismatched CLIPScore with bootstrap CI and permutation p :

Table 5: Classifier-free guidance sweep: paired prompt-alignment gap (matched minus mismatched CLIPScore). $n=16$ held-out prompts.

CFG w	paired alignment gap	CI95	perm p
1.0	+0.0125	$[-0.0037, +0.0282]$	0.026
2.0	+0.0407	$[+0.0222, +0.0588]$	≤ 0.001
3.0	+0.0500	$[+0.0309, +0.0677]$	≤ 0.001
5.0 (shipped)	+0.0471	$[+0.0286, +0.0665]$	≤ 0.001
7.5	+0.0452	$[+0.0214, +0.0715]$	≤ 0.001

Three conclusions, one against us. (1) **Turning CFG on was correct and now has real evidence:** the $w=1$ and $w=5$ CIs do not overlap (5.0’s lower bound +0.0286 exceeds 1.0’s upper bound +0.0282), and at $w=1$ the CI includes values indistinguishable from ignoring the prompt. (2) $w=3.0$ **is the optimum, not $w=5.0$** , on the point estimate — but the CIs for 2.0 through 7.5 overlap heavily; this is a plateau that $n=16$ cannot resolve, so the shipped value is not wrong, it is arbitrary within a flat region. (3) **This measures alignment, not fidelity:** guidance trades realism for adherence at high w and nothing here measures that trade.

Corrections (kept). Guidance is mostly a *beautification* knob, not a comprehension one: from $w=1 \rightarrow 7.5$ saturation rises +70% while sharpness is flat (0.0527 $\rightarrow$ 0.0533) and the subject does not change — the car is already a car at $w=1.0$. $w=5.0$ overshoots the data (saturation 0.317 versus ground-truth 0.186) and $w=7.5$ degrades structure. And the shipped value’s provenance is honest: $w=5.0$ went to the whole gallery on an $n=16$ saturation screen that was *not* significant (1 $\rightarrow$ 3 hits, exact binomial $p=0.074$); the alignment evidence above was found *after* the fact, which is exactly what the audit objected to. The promised $n=64$ sweep is still owed, with a now-narrower job: choose between $w \approx 3$ and $w \approx 5$ inside a known plateau and measure the fidelity side of the trade. The point stands confidently: guidance amplifies the scene the prompt already picked; it does not make the prompt pick a better subject (Figure 2).

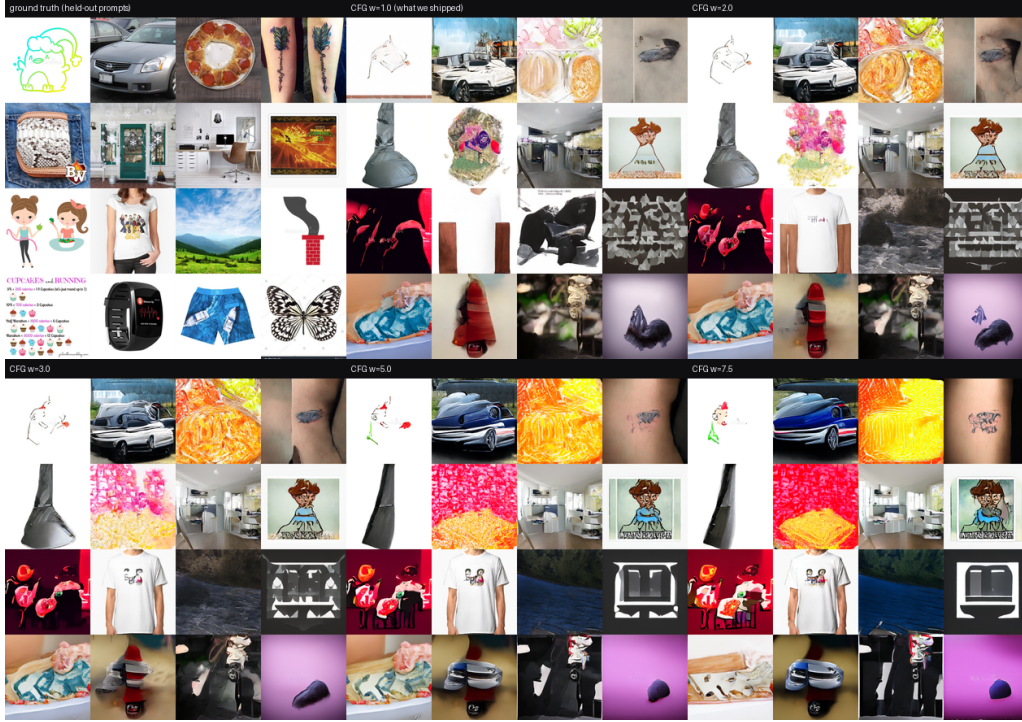


Figure 2: The classifier-free guidance sweep on 16 held-out prompts. Prompt alignment rises from $w=1$ to $w=3$ and then plateaus; saturation rises +70% across the range while sharpness stays flat. $w=3$ is the alignment optimum. The effect is real but mostly beautification, not comprehension.

7 A small, real prompt effect

The clearest positive conditioning result comes from a held-out test designed under the audit’s corrections. We draw $n=64$ prompts seeded-at-random (seed 7) from the held-out eval split, captions receipted and never seen in training, verified isolated by an engineering audit. We sample 64 images from noise (50 Euler steps, seed 42) and ask how often each sample’s nearest ground-truth-in-latent-space is its *own* caption’s image.

The result is 4 of 64 = 6.2% against a chance baseline of $1/64 = 1.6\%$. The exact binomial $P(X \geq 4 \mid n=64, p=1/64) = 0.018$ — significant at 0.05. The honest read is that the conditioning signal is **real but weak**: four-times chance on a coarse latent-distance instrument, with no quality claim attached. The qualitative record is the held-out grid of Figure 1, and its matched ground truth in Figure 3 makes the gap concrete.

8 What we retracted, and the honest limits

This section is a feature, not an apology. Keeping overturned claims visible is part of the measurement discipline, and each correction below is already stated in context above; Table 6 collects them.

The injection bakeoff deserves a line of its own because it is clean. Arms were step-aligned (7,900 steps each), same seed, batch-identical until $0.6 \cdot T$, after which arm B injected the top-10% aesthetic subset. B was worse on the primary held-out MSE (0.18226 vs 0.17879, +0.00347) *and* on the quality-slice metric designed to favour it (0.20651 vs 0.20556). Injection demonstrably ran (6,320 post-threshold draws), so this is a real test of the hypothesis rather than a null from a lever that never fired, and the direction is consistent at every eval after injection starts. The honest bound: $n=1$ per arm, no “ $\times$ seed-noise” multiplier quoted; for scale only, the

Table 6: Claims made during the stage and corrected under adversarial self-audit.

claim made	correction
“LR is insensitive in this range; the spec default stands”	Violated the pre-registered rule ($0.00042 > 0.00030$); restated to “muon-lr 0.01 was the screening winner.” The “ $\sim 1.4\times$ ” softening was post-hoc.
“ $2\times$ is worse by $3.6\text{--}5.6\times$ seed noise”	Multipliers withdrawn ($n=1$ pair; CI on the bound spans $[0.45\times, 30\times]$). What stands: 0.04 worst at all 8 eval points, and stable.
“Optimiser isolated; Muon beats AdamW by $9.2\times$ seed noise”	Not isolated (trunk WD 0.0 vs 0.1 confound on $\sim 85\%$ of params); multiplier withdrawn; “recipe locked” downgraded to “provisional pending AdamW- $3e-3$.” Directional lead survives.
“The sampler default stays 1.0 until an $n=64$ CFG sweep picks w ”	That sweep was never run; $w=5.0$ shipped on a non-significant $n=16$ screen ($p=0.074$). $n=64$ sweep still owed.
“CFG improves comprehension”	Mostly beautification: +70% saturation, flat sharpness, subject unchanged.
“Every arm clearly produces a different picture”	Three of six arm pairs are as similar as a seed change; the optimiser verdict is perceptually ambiguous.
“Quality-injection helps”	Injection <i>hurt</i> : 0.18226 vs 0.17879 primary (+0.00347), and +0.00095 on the quality-slice it was designed to favour.
“micro-batch 64 at 150M fits” (spec)	Refuted as written: DiT-B OOMs at mb32 (23.4 GB) without grad-checkpointing.

observed gap is an order of magnitude above both the 6 h seed-pair bound (0.00003) and the 90 min one (0.00030). A plausible, untested mechanism is that pure-subset decay collapses data diversity in the last 40% of training. The next falsifier, if revisited, is a 0.25-fraction mixture on a diversity-matched subset.

9 What is unresolved

1. **Does M1 clear its own gate?** The gate is a judge-protocol 2AFC win and a CD-FVD/JEDi comparison against an SD1.5-class baseline at matched compute. The pipeline is validated and the arms are run; the judge-and-metric scorecard against the baseline is the remaining receipt, and until it exists M1 is not *declared* — the artifact is the truth, not the report.
2. **The optimiser verdict needs power.** Muon leads on loss, but the perceptual comparison straddles the floor. The queued AdamW-3e-3 falsifier, a trunk-WD-matched Muon variant, and ≥ 3 -seed re-evaluation of all final checkpoints are what would settle it.
3. **The $n=64$ CFG sweep.** Pick w between ~ 3 and ~ 5 inside the known plateau and, for the first time, measure the alignment-versus-fidelity trade rather than alignment alone.
4. **Why does the prompt effect sit at 4/64?** It is real ($p=0.018$) and weak. Whether it is corpus, text-encoder choice, or undertraining that caps it is exactly what the later bakeoffs are designed to isolate.
5. **The under-powered measurement floor itself.** The seed floor rests on two disagreeing pairs; 64-prompt grids and ≥ 4 seed pairs would turn the arm-similarity screen into something that can actually rank interventions by their effect on the picture.

M1 is a confident result with a sharp lesson. The pipeline works: a diffusion transformer trained from scratch on two consumer cards learns to draw recognisable scenes and to prefer its own prompt, provably and reproducibly. And the measurement discipline built around it is strong enough to tell us something uncomfortable and useful — that most of what we tuned moved a number without moving a picture. That is the knowledge the video rungs of the program will be built on: demand both the loss and the picture, and never confuse one for the other.

References

- [1] W. Peebles and S. Xie. Scalable Diffusion Models with Transformers. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.
- [2] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [3] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2210.02747.
- [4] X. Liu, C. Gong, and Q. Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations (ICLR)*, 2023.

- [5] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, et al. Scaling Rectified Flow Transformers for High-Resolution Image Synthesis. In *International Conference on Machine Learning (ICML)*, 2024.
- [6] J. Ho and T. Salimans. Classifier-Free Diffusion Guidance. arXiv:2207.12598, 2022.
- [7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021.
- [8] S. Changpinyo, P. Sharma, N. Ding, and R. Soricut. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training to Recognize Long-Tail Visual Concepts. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [9] K. Jordan, Y. Jin, V. Boza, J. You, F. Cesista, L. Newhouse, and J. Bernstein. Muon: An Optimizer for Hidden Layers in Neural Networks. Technical note, 2024. <https://kellerjordan.github.io/posts/muon/>.
- [10] A. Yang et al. (Qwen Team). Qwen3 Technical Report. arXiv:2505.09388, 2025.
- [11] Wan Team (A. Wang et al.). Wan: Open and Advanced Large-Scale Video Generative Models. arXiv:2503.20314, 2025.
- [12] W. Kong et al. (Tencent Hunyuan). HunyuanVideo: A Systematic Framework for Large Video Generative Models. arXiv:2412.03603, 2024.
- [13] J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [14] S. Hu, Y. Tu, X. Han, C. He, G. Cui, X. Long, et al. MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies. arXiv:2404.06395, 2024.