RESEARCH · APPLE METAL · AGAINST THE FIELD

Exe AI on Apple Silicon: Video 7.7× Faster Than the Original, Ahead of MLX on a Newer Chip

LTX-2.3 and MiniMax H3 on an M2 Max beside the original stable-diffusion.cpp and every published figure for the same models · each bar carries its machine, its settings and what its time counts

Exe — Exe AI, developer studio

Exe AI measured on Apple M2 Max (38-core GPU, 64 GB unified memory) · all other bars from their sources, Section 4

7.7×
LTX-2.3 whole video, 1280×704, 121 frames with audio, against the original stable-diffusion.cpp on the same Mac: 2,715.0 s → 350.8 s

111×
its video decoder alone: 2,278.0 s → 20.6 s

1.11×
LTX-2.3 denoising per megapixel ahead of Apple's MLX on an M4 Max, two chip generations newer, both with audio

33B
MiniMax H3 with all its weights at 8-bit, whole run in 42.35 GB on the Mac; on Nvidia even the pruned build peaks at 74.9 GB in BF16, more than any consumer card

Bars: Exe AI · published figure · original stable-diffusion.cpp · shorter is faster

1 Against the original

LTX-2.3 22B Distilled · whole run with audio, original against Exe AI

Lightricks · 1280×704 · 121 frames · 4 steps · from the start of the process to the written video file, loading included

Original sd.cpp · M2 Max 38-core · unmodified, Q8_0 · the same command — 2,715 s

Exe AI · M2 Max 38-core · sd.cpp with own Metal kernels, Q8_0 — **351 s**

Video decoder alone 2,278.0 s → 20.6 s (111×); denoising 384.6 s → 282.7 s (1.36×). The two videos are not bit-identical: the Exe AI video against the original's has a PSNR of 32.95 dB on average over the 121 frames, 31.34 dB at the lowest frame.

MiniMax H3 · whole run with audio, original against Exe AI

MiniMax · 864×480 · 56 frames · 20 steps · all 33B weights at Q8_0 · loading included

Original sd.cpp · M2 Max 38-core · unmodified · the same command — 1,247 s

Exe AI · M2 Max 38-core · sd.cpp with own Metal kernels · peak memory 42.35 GB — **920 s**

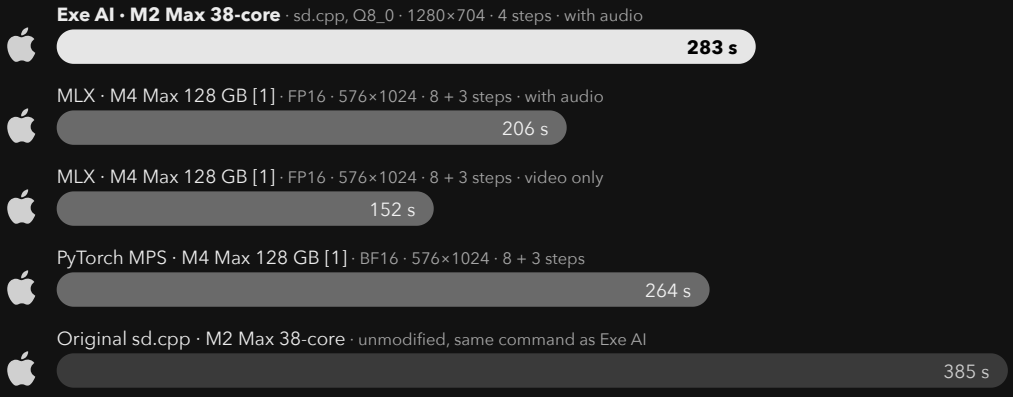
1.36× faster on the whole run; denoising 1,128.2 s → 821.5 s. Not bit-identical either: PSNR 38.00 dB on average over the 56 frames, 35.92 dB at the lowest frame.

The original is stable-diffusion.cpp as published, the Exe AI build is its fork with own Metal kernels; both ran the same command on the same Mac. Report 6 has every model family; the rest of this report sets the Exe AI times beside other software and other machines.

2 On Apple Silicon

LTX-2.3 22B Distilled · denoising on Apple Silicon

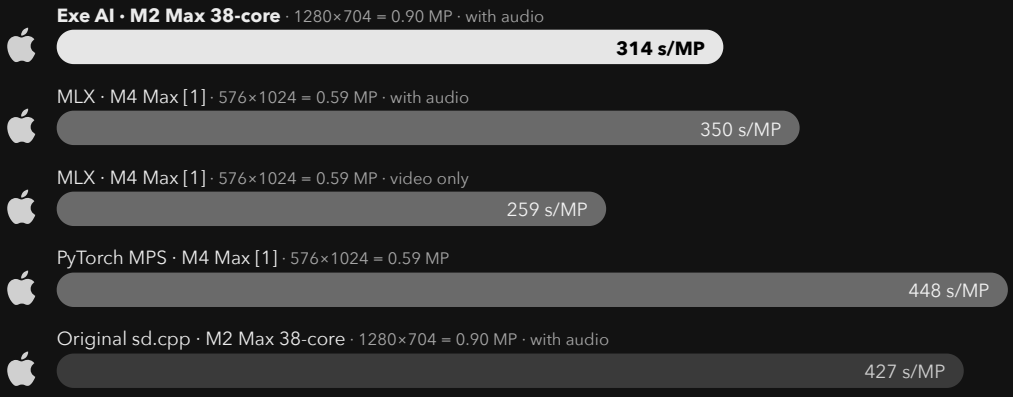
Lightricks · 121 frames · time of the denoising steps only, decoders and loading left out



Exe AI draws 1.53× the pixels of each frame of the M4 Max runs, on a chip two generations older. The 8 + 3 schedule runs eight steps at half resolution and three at full resolution; Exe AI runs four at full resolution.

LTX-2.3 22B Distilled · denoising per megapixel of frame

the same five runs, each divided by the pixels of one frame in millions · same 121 frames



Dividing by the pixel count is kind to the smaller frame: attention grows faster than the pixel count, so the larger Exe AI frame is the harder one per pixel. The step schedules still differ, as above.

No published figure names stable-diffusion.cpp on a Mac for these models, and none names an M2 Max or an M3 Max with LTX-2.3; the M4 Max runs of source [1] are the only like-for-like Apple figures found. They give the same 121 frames of the same distilled model, measured on one machine against PyTorch.

Other published Apple figures

MODEL, SETTINGS	PUBLISHED FIGURE	TIME	EXE AI, M2 MAX	RATIO	WHAT DIFFERS
LTX, 1280×704, 121 frames, with audio	Phosphene (MLX), M2 Max 64 GB: median of 37 renders on 4 machines, loading included; 253 renders on 6 M2 Max of every memory size give the same median [15]	360 s	350.8 s · LTX-2.3, whole run	–	LTX-2.5 instead of 2.3, 4-bit instead of Q8_0; the step plan is not in the telemetry; 24 instead of 25 frames per second
MiniMax H3, about 0.41 MP per frame	h3.c, M2 Max 32 GB: denoising of 480×864, 124 frames, 8 passes, BF16 weights streamed from the SSD [16]	1,694.5 s	821.5 s · denoising, 864×480, 56 frames, 20 steps	–	2.2× the frames, 8 instead of 20 passes, 32 GB with streamed weights; the run's video had a fault, its time is reported

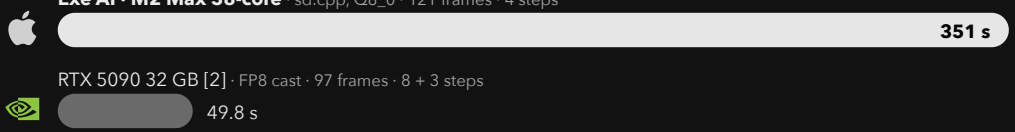
MODEL, SETTINGS	PUBLISHED FIGURE	TIME	EXE AI, M2 MAX	RATIO	WHAT DIFFERS
MiniMax H3, video with audio	Phosphene, M2 Max of every memory size: median of 11 renders on 3 machines, 768×448, 124 frames, 8 passes, loading included [15]	1,070 s	920.3 s · whole run, 864×480, 56 frames, 20 steps	–	other frame size, 124 against 56 frames, 8 against 20 steps, BF16 or Q8 by memory size
SDXL, 1024², 20 steps, with guidance	Apple Core ML (CPU and GPU), M2 Max, GPU cores not stated: end to end, median of three runs [17]	37 s	43.9 s · whole run, CyberRealistic XL, CFG 5	0.84×	same chip; Apple does not say whether loading is inside its time, the Exe AI time includes it. Denoising alone: 0.57 it/s = 35.1 s against 39.0 s (0.90×). SDXL base against a fine-tune of the same network
Krea 2 Raw, 1024², 52 steps, CFG 3.5	ComfyUI, PyTorch MPS, M1 Max 64 GB: one image [18]	2,822 s	1,258.6 s · whole run	2.24×	M1 Max, one generation older; text encoder at FP8 there
MOSS-Image Base, 1024², 50 steps, CFG 3	diffusers, PyTorch MPS, M1 Max 64 GB: pipeline call without loading [19]	1,189.9 s	513.2 s · whole run with loading	2.32×	M1 Max, one generation older; the published time leaves loading out
Iris 3B, 1024², 100 steps, guidance 3	own Metal port, M5 24 GB: first image with loading, 457.5 s warm [20]	464.9 s	893.7 s · whole run	0.52×	M5, three generations newer, with neural accelerators in its GPU

- Ratio: published time ÷ Exe AI time; above 1 Exe AI is faster. Given only where model, frame size, steps and guidance match; the chip may differ.
- Against Apple's own SDXL: on the same chip Exe AI is slower, 0.84× end to end and 0.90× in the denoising alone.
- LTX on an M2 Max: the telemetry for LTX-2.5 at the same frame size and length lands at the same order of time as Exe AI with LTX-2.3; model version, precision and step plan differ, so no ratio is given.

3 Against Nvidia

LTX-2.3 22B Distilled · whole run with audio, Apple against Nvidia

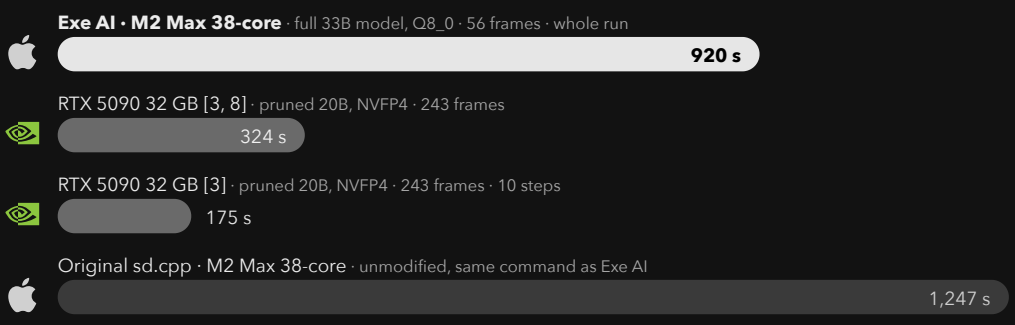
Lightricks · 1280×704 · from the start to the written video file



Per frame: Exe AI 2.90 s, the RTX 5090 0.51 s. The original stable-diffusion.cpp needs 2,715 s for the Exe AI command; its bar is left out so the two short bars stay readable.

MiniMax H3 · 864×480, Apple against Nvidia

MiniMax · video with audio · 20 steps unless noted



The RTX 5090 draws 4.3× the frames with a build that keeps 61 % of the weights, at 4-bit precision; the source does not say whether loading and decoding are counted. Per frame at 20 steps: Exe AI 16.4 s, the RTX 5090 1.33 s.

- LTX-2.3, per frame: the RTX 5090 is 5.6× faster than Exe AI [2]. What differs: FP8 against Q8_0; 97 against 121 frames; eight steps at half resolution and three at full resolution against four at full resolution; the RTX 5090 time starts at the pipeline call, the Exe AI time at the start of the process, model loading included.
- MiniMax H3, per frame: the RTX 5090 is 12× faster at 20 steps [3, 8]. Reservations: the article gives the 324 s without frame size or length; the author's model card gives them at the same resolution and seed as its measured task, 864×480 with 243 frames, but does not write 864×480 next to them. The run used 4-bit NVFP4 weights and an NVFP4 text encoder on torch 2.11.0+cu128, where ComfyUI registers no FP4 matrix kernel and falls back to PyTorch's eager path [9]. Whether loading and decoding are inside the 324 s is not stated.
- Pruning: MiniMax states that about 13B of the 33B weights sit in AdaLN branches that inference does not need to load [10]. The pruned build of the RTX 5090 therefore keeps what the video needs; what it gives up is precision, 4-bit against the 8-bit of Exe AI.
- MiniMax H3 on an RTX 3090 (24 GB): pruned INT8, 864×480, 124 frames, 20 steps, 9.00 s per step, about 180 s for the sampling alone [12]. Exe AI samples its 56 frames in 821.5 s. Per frame of denoising: the RTX 3090 1.45 s, Exe AI 14.67 s – the RTX 3090 is about 10× faster, on more than twice the frames.

What the two models need on Nvidia

MODEL	BUILD	MEMORY	MACHINE	SOURCE
LTX-2.3 22B	Lightricks' ComfyUI nodes, stated requirement	32 GB or more	of the GeForce cards only the RTX 5090	[13], [7]
LTX-2.3 22B	transformer file alone, BF16 / FP8 / INT8	42.02 / 23.47 / 21.51 GB	BF16 above every GeForce card	[14]
LTX-2.3 22B	Exe AI: Q8_0 with the Gemma 3 12B text encoder, whole run	31.67 GB peak	M2 Max, 64 GB unified memory	night run
MiniMax H3 33B	the maker's deployment, BF16	four GPUs	–	[10]
MiniMax H3, pruned 20B	BF16 with an INT8 text encoder; reference-to-video, 480×854, 124 frames, 20 steps	71,437 MiB = 74.9 GB peak	RTX PRO 6000, 96 GB	[11]
MiniMax H3, pruned 20B	NVFP4 with an NVFP4 text encoder / INT8; 864×480, 243 frames	26,914 / 28,581 MiB peak	RTX 5090, 32 GB	[3], [8]
MiniMax H3 33B	Exe AI: all weights at Q8_0, whole run	42.35 GB peak	M2 Max, 64 GB unified memory	night run

- Units: GB = 10^9 bytes, MiB = 2^{20} bytes; 26,914 and 28,581 MiB are 28.2 and 30.0 GB.
- Exe AI peaks: the peak memory footprint of the whole process, in unified memory that the GPU shares with the system; a card's figure counts the memory of the card.
- Smaller cards: no LTX-2.3 run with complete settings was found for a card with 24 GB or less.

Image models on Nvidia, for scale

MODEL, SETTINGS	PUBLISHED, NVIDIA, MODEL LOADED	EXE AI, M2 MAX, LOADING INCLUDED
SDXL 1.0, 1024^2, 20 steps, one image	ComfyUI default workflow (Euler, CFG 8), second run [21]: RTX 3060 12 GB 14.92 s · RTX 4090 3.08 s · RTX 5090 2.44 s	whole run 43.9 s, denoising 39.0 s (CyberRealistic XL, Euler, CFG 5)
Qwen-Image 2.1, 1024^2	40 steps without guidance; RTX 4090 24 GB, diffusers BF16 with the model offloaded to the CPU, a few server requests: about 45 s [22]	30 steps with CFG 3, two passes per step: whole run 479.7 s, denoising 443.1 s

- No ratio: the published times are of a loaded model, the Exe AI times include loading; the Qwen-Image 2.1 figure runs without guidance, so half the passes per step.

4 How the figures were made

Exe AI	Release `master-1000-82a37b3`, night-run command (`nightrun/run_matrix.py`): seed 42, one prompt, `--flash-attn on --params-backend te=disk`; times and peak memory from `results.tsv`, columns `sampling_s` (denoising), `decode_s` (video decoder), `wall_s` (whole run) and `peak_footprint_b`; PSNR against the original from `compare.tsv`
Original	stable-diffusion.cpp `master-1000-95ff762`, unmodified, the same command
Published	copied from the source in brackets, with the settings the source gives; nothing was measured again
Scale	one scale per figure, the longest bar of the figure full width
Warm and cold	every Exe AI time includes loading the model; most published Nvidia times are of a loaded model or start at the pipeline call

- What differs, LTX-2.3: Exe AI runs the distilled model at Q8_0 in four steps at full resolution; the published runs use eight steps at half resolution, an upscaler and three steps at full resolution, at FP16, BF16 or FP8. The frame sizes differ (1280×704 against 576×1024), the RTX 5090 run has 97 frames instead of 121.
- What differs, MiniMax H3: the RTX 5090 runs a build pruned from 33.1 to 20.1 billion weights, at 4-bit NVFP4, with 243 frames against 56; the source does not state whether loading and decoding are inside its time. The Exe AI bar is the whole run.
- What the cards do not show: the per-step gains of the current release `07f8c72` (MiniMax H3 −13 %, LTX-2.3 −10 %, Report 5), which has no whole video runs yet.
- Found but not drawn: LTX-2.5 with MLX on an M3 Max, 768×512, five seconds in 5–8 minutes [5] – another model version, a range; MiniMax H3 on an RTX 5060 Ti 16 GB, 896×672, six seconds in 9 min 55 s [4] – steps, frames and precision not given; LTX-2.3 dev at FP8 on an RTX 5090, 481 frames at 1280×720 in 82 s [6] – a community report on the larger dev model, most of the time loading by its author's account.
- Maker marks: the vendor vectors of the Exe AI app, unchanged; all marks belong to their owners.

Sources

- [1] `huggingface.co/gajesh/LTX-2.3-mlx-fp16` – LTX-2.3 distilled on an M4 Max 128 GB, MLX against PyTorch MPS, same machine and settings
- [2] `huggingface.co/datasets/witcheer/rtx-5090-benchmarks (reports/ltx-2.3.md)` – LTX-2.3 distilled with audio on one RTX 5090, timed from the pipeline call to the encoded file
- [3] `ai-muninn.com/en/blog/minimax-h3-nvfp4-rtx5090` – MiniMax H3 on one RTX 5090, NVFP4, pruned build, ComfyUI
- [4] `virse.ai/blog/minimax-h3-guide` – MiniMax H3 on an RTX 5060 Ti 16 GB, settings not given
- [5] `gist.github.com/DanielHauschildt (LTX-2.5 on Apple Silicon)` – LTX-2.5 with MLX on an M3 Max 64 GB
- [6] `huggingface.co/Lightricks/LTX-2.3/discussions/16` – LTX-2.3 dev FP8 on an RTX 5090, community report
- [7] `smeltcore.com/recipes/ltx-video-2-3-on-rtx-5090-first-card-that-hits-the-32-gb-floor` – memory LTX-2.3 needs on Nvidia cards
- [8] `huggingface.co/coolthor/MiniMax-H3-pruned-NVFP4` – model card of the author of [3]: 324 s → 172 s from 20 to 10 steps at "same resolution and seed"; measured task 864×480, 243 frames, seed 16; RTX 5090, ComfyUI 0.30.1, torch 2.11.0+cu128, 500 W limit
- [9] `ai-muninn.com/en/blog/minimax-h3-rtx5090-config-stack` – follow-up of [3]: on torch cu128 ComfyUI registers no NVFP4 matrix kernel and the FP4 products fall back to eager code
- [10] `huggingface.co/MiniMaxAI/MiniMax-H3` – the maker's card: BF16 checkpoints, served on four GPUs; about 13B parameters in AdaLN branches that inference-only deployment does not need to load
- [11] `github.com/DominikaBest/minimax-h3-quality-comparison` – MiniMax H3 pruned BF16 with an INT8 text encoder on an RTX PRO 6000 96 GB, sampled peak memory
- [12] `github.com/Comfy-Org/ComfyUI/issues/15263` – MiniMax H3 pruned INT8 on an RTX 3090, ComfyUI 0.30.1, sampling time per step

- [13] github.com/Lightricks/ComfyUI-LTXVideo – Lightricks' LTX nodes for ComfyUI: "CUDA-compatible GPU with 32GB+ VRAM"
- [14] huggingface.co/Kijai/LTX2.3_comfy – LTX-2.3 transformer files for ComfyUI in BF16, FP8 and INT8, file sizes
- [15] phosphene.video/benchmarks – render telemetry of the Phosphene app (MLX) per chip, medians, model loading included, queue time left out
- [16] github.com/antirez/h3.c/issues/72 – MiniMax H3 with h3.c on an M2 Max 32 GB, timings of text encoder, denoising and decoder
- [17] github.com/apple/ml-stable-diffusion – Apple's Core ML Stable Diffusion: SDXL base 1.0 at 1024², 20 steps, end to end as the median of three runs
- [18] lilting.ch/en/articles/krea-2-m1-max-comfyui – Krea 2 Raw in ComfyUI with PyTorch MPS on an M1 Max 64 GB
- [19] lilting.ch/en/articles/moss-image-base-m1-max-local – MOSS-Image Base with diffusers and PyTorch MPS on an M1 Max 64 GB
- [20] github.com/TKY-27/iris-3b-metal – Iris 3B with its own Metal port on an Apple M5 24 GB
- [21] github.com/comfyanonymous/ComfyUI/discussions/2970 – SDXL 1.0 in the ComfyUI default workflow on Nvidia and AMD cards, second run
- [22] github.com/MindrLabs/image-generation-qwen-image-2.1 – Qwen-Image 2.1 BF16 with diffusers on an RTX 4090 24 GB, model offloaded to the CPU

EXE AI · DEVELOPER STUDIO