

Qwen3.5-9B LoRA fine-tuning on Apple's WWDC26 recipe: one MacBook Pro M5 Max with NukeTorch

Klay Moon, NukeTorch · Report v1.1 · Apple M5 Max, 48 GB unified memory, macOS 26.7.1 · GB = 2³⁰ bytes

Headline

- **1,440.8 trained tokens per second** over 1,000 steps (3 passes, median) on one Apple M5 Max (48 GB): 8.00× the ~180 Apple reported for one M3 Ultra Mac Studio and 2.40× the ~600 for four (Apple's figures, on its machines).
- **Four Macs:** the four-Mac pipeline report's 5,358.5 trained tokens per second is 93.0% of four times this single-Mac rate.
- **One Mac and four Macs:** both run 1,000 steps of Apple's recipe. On one Mac it trains 4 examples per step, Apple's one-Mac setting; on four Macs, 16 per step (4 per Mac). The four-Mac job does not fit in one 48 GB Mac. Speeds are compared in tokens per second.
- **Energy: 93.4 mJ per trained token** (the whole machine's consumption; Table 2).
- **MLX as shipped does not run this recipe on this Mac:** it stops with "[METAL] Command buffer execution failed: Insufficient Memory" before its first step. On the Apple M4 Max (128 GB) its peak memory footprint for the same recipe is 57.4 GB, more than this Mac's 48 GB.
- **Same training:** the same 70 batches as MLX's own run, and every 10-step window's loss within 0.91% of MLX's.
- **Peak memory:** 28.3 GB on a 48 GB Mac (Table 4).

Table 1. Speed

Source	Machine	Run length	Trained tokens/s
Apple, as published (WWDC26)	1 M3 Ultra Mac Studio	not stated	~180
Apple, as published (WWDC26)	4 M3 Ultra Mac Studios	not stated	~600
MLX 0.32.4 / mlx-lm 0.32.2, as shipped	1 Apple M4 Max (128 GB)	70 steps (stopped by the recorder)	159.2
MLX 0.32.4 / mlx-lm 0.32.2, as shipped	1 Apple M5 Max (48 GB)	stopped before step 1	out of memory
NukeTorch	1 Apple M5 Max (48 GB)	1,000 steps × 3 passes (median)	1,440.8

Apple's figures are spoken approximations from its session, on its machines: "Single M3 Ultra is processing around 180 tokens per second, while on the cluster we process around 600 tokens per second" ([WWDC26 session 233](#), ["Explore distributed inference and training with MLX", chapter "Distributed fine-tuning" \(figure at 18:10\)](#)). Its single-machine command is the recipe here: `mlx_lm.lora --model "Qwen/Qwen3.5-9B" --data "mlx-community/wikisql" --train --batch-size 4`. The ratios against Apple compare different machines; MLX on the Apple M5 Max produced no rate.

Table 2. The 1,000-step runs (365,328 trained tokens each)

Pass	Trained tokens/s	mJ per token	Whole machine (W)
1	1,439.3	92.9	133.7
2	1,440.8	93.5	134.7
3	1,444.7	93.4	134.9
Median	1,440.8	93.4	

Energy per token = the machine's consumption (the battery controller's load counter, averaged between its refreshes inside the run) × the run's training time ÷ its tokens. The battery was finishing its charge during 3 of the 3 runs, so their power includes some charging.

Table 3. The same training as MLX: loss, window by window (70 steps)

Steps	NukeTorch	MLX as shipped	Difference
1–10	1.739	1.755	-0.89%
11–20	1.232	1.237	-0.38%
21–30	1.155	1.157	-0.16%
31–40	0.967	0.965	+0.24%
41–50	1.098	1.100	-0.16%
51–60	1.086	1.087	-0.06%
61–70	0.890	0.892	-0.24%

The same batches in MLX's order; NukeTorch's running token count equals MLX's printed count at every window.

Table 4. Memory (peak process footprint, GB = 2³⁰ bytes)

Source	Machine	Peak	Outcome
MLX 0.32.4 / mlx-lm 0.32.2, as shipped	Apple M4 Max	57.4	ran
MLX 0.32.4 / mlx-lm 0.32.2, as shipped	Apple M5 Max	37.5	out of memory
NukeTorch	Apple M5 Max	28.3	all steps

How to reproduce

1. **Machine:** one MacBook Pro 16-inch, Apple M5 Max (18-core CPU, 40-core GPU, 48 GB), macOS 26.7.1, on mains power.
2. **Cooling (declared):** no external fans; the Mac's own fans under automatic control; the same as the four-Mac runs.
3. **Model and data:** Qwen/Qwen3.5-9B at revision c20223623576; mlx-community/wikisql at revision 886acf6d49be, in MLX's batch order.

4. **Recipe:** Apple's single-machine command, unchanged: batch 4, LoRA rank 8, scale 20, layers 16–31, Adam at $1e-5$; 1,000 steps per run, three runs, 60 seconds idle between runs.
5. **Engine:** NukeTorch build d30de70f from a clean tree, built fresh (available to licensees); a quiet-machine check before every run.
6. **Measure:** trained tokens $\div$ the sum of step times, step 0 included.

Prepared by NukeTorch Inc. Report v1.1. Author-run measurements; no independent reproduction yet.