Hybrid Conv+GQA vs. Dense Transformer: Efficiency, Long-Context Behavior, and LoRA Adaptation

Zhaokun Wang
Institute of Computational Linguistics
Heidelberg University, Heidelberg, Germany
zhaokun.wang@stud.uni-heidelberg.de

Abstract

We present a comparison between LiquidAI's hybrid Conv+GQA language model (**LFM2.5-1.2B-Instruct**) and a dense-transformer baseline (**Qwen3-1.7B**) across inference efficiency, long-context capability, and LoRA adaptation in this course project. Our evaluation reveals a distinct efficiency–capability trade-off. LFM2.5 demonstrates substantial deployment advantages, delivering up to **11.5**× higher throughput, faster time-to-first-token, and a **44–47%** reduction in both inference and training memory footprint. Conversely, while both models achieve perfect single-needle retrieval, Qwen3 proves significantly more robust on harder capability metrics, outperforming the hybrid model on multi-needle retrieval, exact associative recall, and absolute post-fine-tuning quality. We conclude that the hybrid Conv+GQA architecture provides a highly efficient operating point for resource-constrained deployment, whereas the dense baseline remains the safer choice for tasks demanding precise long-range retrieval and high-fidelity adaptation.[1]

1 Introduction

Transformer-based language models have become the dominant paradigm in NLP (1), but their quadratic attention cost remains a persistent bottleneck for long-context inference and efficient deployment. This limitation has motivated a growing line of work on more efficient sequence architectures, including state-space models such as Mamba (11), efficient Transformer variants such as Reformer and Longformer (4; 2), and recurrent or memory-augmented alternatives such as xLSTM and Titans (5; 6), as well as hybrid designs that combine attention with cheaper sequence-mixing mechanisms.

Liquid AI's **LFM 2.5** is treated here as a released hybrid checkpoint used in our experiments. We therefore frame our claims about this model empirically under our evaluation protocol, rather than attributing them to a separate architecture paper. In our study setup, the model is analyzed as a Conv+GQA hybrid: it contains 16 blocks, of which 10 are double-gated convolution blocks and 6 use Grouped-Query Attention (GQA) (12). The architectural premise we test is that full attention may not be necessary at every layer: local dependencies can be handled by convolutional operators with lower computational cost, while a smaller number of attention layers can be reserved for global information routing.

This paper studies that premise through an empirical comparison between **LFM2.5-1.2B-Instruct** and the dense-transformer baseline **Qwen3-1.7B** (13). We focus on four practical dimensions:

1. **Baseline capability**: long-context retrieval and perplexity behavior;
2. **Inference efficiency**: throughput, latency, and memory across input lengths and hardware settings;
3. **LoRA adaptation**: fine-tuning resource use and post-fine-tuning quality;
4. **Supporting mechanistic probes**: targeted analyses used to interpret the observed trade-offs.

Our central question is whether a hybrid Conv+GQA architecture can provide a better deployment-oriented operating point—improving efficiency and memory behavior without giving up too much long-context capability relative to a dense Transformer. In particular, we examine whether the efficiency gains of the hybrid model come with weaker performance on more demanding retrieval and post-adaptation evaluations.

Because the compared checkpoints differ not

[1] Codes are publicly available at https://github.com/BufferHund/ATTProject

Table 1: Summary of models compared in this study.

Property	LFM2.5-1.2B-Instruct	Qwen3-1.7B
Developer	Liquid AI	Alibaba Cloud
Parameters	1.2B	1.7B
Architecture	Hybrid: Conv + GQA	Dense Transformer
Attention Layers	6 / 16 (hybrid)	28 / 28 (full)
Context Window	32K tokens	32K tokens

Table 2: Hardware configurations used for evaluation.

Component	Primary (CUDA)	Cross-Platform (Mac)
Hardware	NVIDIA RTX 4060 (8 GB)	Apple Silicon (16 GB unified)
Backend	CUDA 11.8	MPS / CPU
Precision	float16	float32

only in architecture, but also in parameter count and likely training recipe, we do not present this study as a clean architecture-only ablation. Instead, we frame it as an exploratory released-checkpoint comparison under a shared evaluation protocol. Within that framing, the strongest result comes from matched CUDA-side efficiency measurements and the completed robust-scored post-fine-tuning comparison, while Apple `MPS` reruns and mechanistic probes serve primarily as stability checks and interpretive support.

2 Methodology

2.1 Models Under Evaluation

Table 1 summarizes the two models evaluated in this study. Because we compare publicly released checkpoints (`LFM2.5-1.2B-Instruct` and `Qwen3-1.7B` (13)), they naturally differ in parameter count (1.2B vs. 1.7B), pretraining corpora, and alignment recipes. Consequently, this evaluation is structured as a comparative study of two distinct model families under a shared protocol, rather than a strictly controlled architectural ablation. Further discussion of how scale may influence these results is deferred to Section 6.2.

2.2 Hardware Platforms

Evaluations were conducted across two distinct hardware backends to assess cross-platform deployment characteristics (Table 2).

2.3 Experimental Protocol

We summarize the core evaluation protocol below, deferring detailed script-level configurations and reproducibility notes to the Appendix.

- **Inference Efficiency.** Throughput, peak memory, and Time-To-First-Token (TTFT) were measured under matched decoding settings. Input lengths were varied systematically while output length and batch sizes were held constant.
- **Long-Context Capabilities.** Synthetic facts were inserted into filler contexts at controlled positions for Needle-In-A-Haystack (NIAH) retrieval (both single- and multi-needle variants). Long-context perplexity (PPL) was computed on matched text segments across varying context windows.
- **LoRA Fine-Tuning.** We applied LoRA fine-tuning using matched optimizer settings across both models. Efficiency is reported via training resource profiles (time and peak memory).
- **Post-Fine-Tuning Evaluation.** Adaptation quality was audited using a robust-scored, logit-based evaluation on fixed benchmark subsets to ensure stable cross-model comparisons.
- **Mechanistic Probes.** Layer ablations, positional interventions, and sparsity measurements were applied using identical logic across models to observe their internal behavior and sensitivity.

Table 3: Throughput (tokens/s) on RTX 4060.

Input	Batch=1		Ratio	Batch=4		Ratio
	LFM	Qwen		LFM	Qwen	
128	31.6	22.6	1.4×	161.6	63.9	2.5×
512	45.8	15.9	2.9×	132.8	64.4	2.1×
1024	36.3	16.1	2.3×	117.0	62.7	1.9×
2048	42.3	15.6	2.7×	107.5	38.4	2.8×
4096	38.3	17.8	2.2×	72.7	6.3	**11.5**×

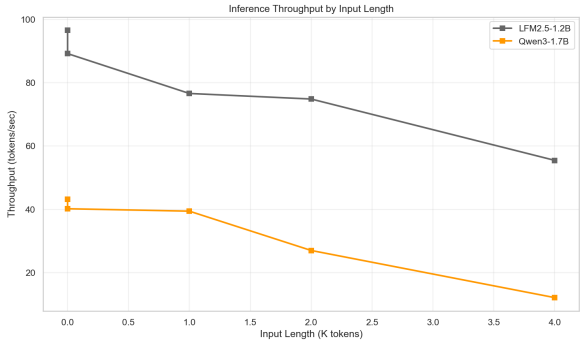
3 Efficiency and Capabilities Evaluation

3.1 Inference Efficiency

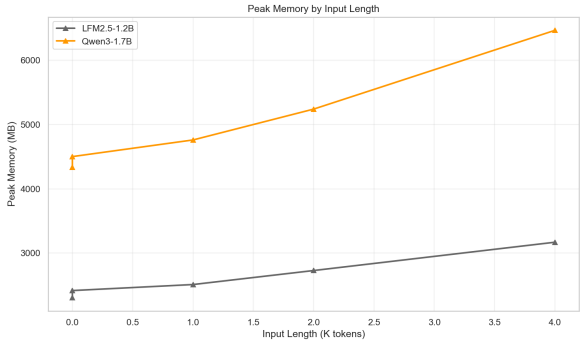
Table 3 and Figure 1 show that the throughput advantage of the hybrid architecture widens significantly under increased load. At a batch size of 4, the ratio grows from 2.5× at 128 tokens to **11.5× at 4096 tokens**. Furthermore, LFM2.5 is highly memory-efficient, requiring only 2,616 MB at 4096 tokens (bs=1) compared to Qwen3's 4,917 MB, representing a **47% reduction** in peak GPU memory.

3.2 Inference Latency

Table 4 and Figure 2 show that latency follows the same qualitative pattern as throughput and memory:



(a) Throughput Scaling



(b) Peak GPU Memory

Figure 1: Inference efficiency comparison on NVIDIA RTX 4060. LFM2.5 maintains a throughput and memory advantage, with the largest gap at longer inputs and higher batch size.

LFM2.5 consistently reaches the first generated token faster, with the clearest advantage around 1K–2K input tokens. This responsiveness is critical for interactive usage, where TTFT is often more perceptible than steady-state throughput.

Beyond TTFT, overall generation latency strongly favors LFM2.5. In a compact decode-latency profile, LFM2.5 achieved a mean decode latency of roughly 32.6 ms compared to 63.5 ms for Qwen3. Similarly, an end-to-end generation evaluation on the `medium_data` checkpoints demonstrated that LFM2.5 averaged 2.47 s total latency versus 8.14 s for Qwen3 under matched prompt-and-output settings. Taken together, these latency metrics uniformly support a substantial deployment-side advantage for the hybrid architecture.

3.3 Cross-Platform Robustness

Figure 3 shows the cross-platform inference results. To verify the stability of our efficiency findings, we replicated inference across Apple Silicon (MPS and CPU backends). LFM2.5's convolutions map efficiently to MPS, maintaining a significant lead over

Table 4: Time to first token (TTFT, ms) on RTX 4060, batch size 1.

Input Length	LFM2.5	Qwen3	Ratio
128	52	68	1.3×
512	123	243	2.0×
1024	144	309	2.1×
2048	209	413	2.0×
4096	364	593	1.6×

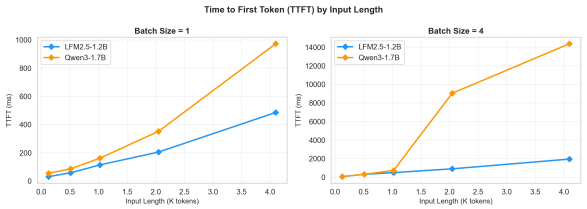


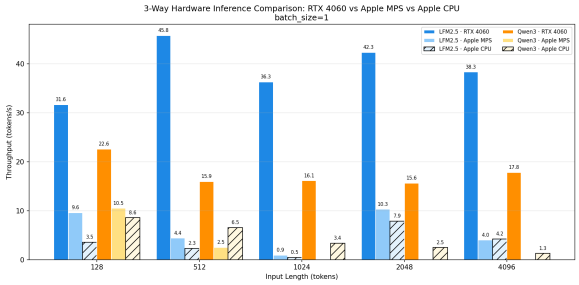
Figure 2: Time-to-first-token comparison on NVIDIA RTX 4060. LFM2.5 reaches the first output token faster across all tested input lengths.

Qwen3. For instance, at 4096 input tokens (batch size 1, 64 output tokens), Qwen3's throughput fell to 0.248 tok/s with a 54.1 s TTFT, whereas LFM2.5 maintained 2.33 tok/s with a 13.3 s TTFT. While native MPS attention kernels for dense models remain less optimized in this environment (causing the CPU backend to occasionally outperform MPS for Qwen3), the relative efficiency advantage of LFM2.5 proves to be highly robust across diverse hardware stacks.

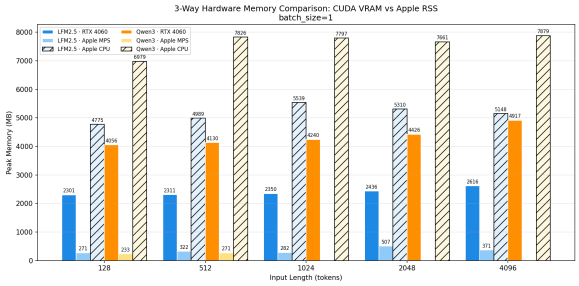
3.4 Needle-in-a-Haystack Retrieval

Figure 4 shows the NIAH results. In the dedicated single-needle benchmark, both models achieve **100% accuracy** across the tested context lengths (1K–16K on CUDA, with matching verification runs up to 4K on MPS).

Table 5 reports a separate, fundamentally harder multi-needle protocol. (Note that the "1 needle" condition here is part of this harder setup and distinct from the dedicated single-needle benchmark). Under this rigorous protocol, Qwen3 maintains perfect accuracy, while LFM2.5 exhibits degradation at higher difficulty settings. This qualitative divergence is hardware-agnostic: replicated MPS runs at 2K and 4K contexts confirm that Qwen3 consistently retains 100% accuracy, whereas LFM2.5 drops to between 66.7% and 96% depending on needle density. These results indicate that while the hybrid model is highly capable of isolated fact retrieval, the dense attention mechanism is signifi-



(a) 3-Way Throughput



(b) Cross-Platform Memory

Figure 3: CUDA vs. Apple MPS vs. CPU inference profiles. The relative efficiency advantage of LFM2.5 remains consistent across platforms.

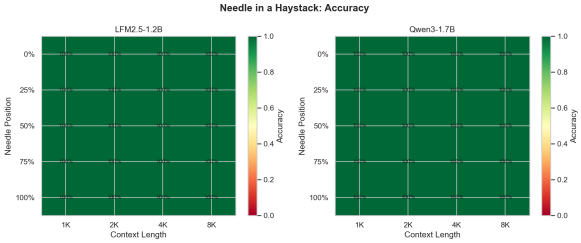
Table 5: Multi-Needle NIAH accuracy (%).

Needles	Context	LFM2.5	Qwen3
1	2K	100	100
3	2K	100	100
5	2K	92	100
1	4K	80	100
3	4K	93	100
5	4K	96	100

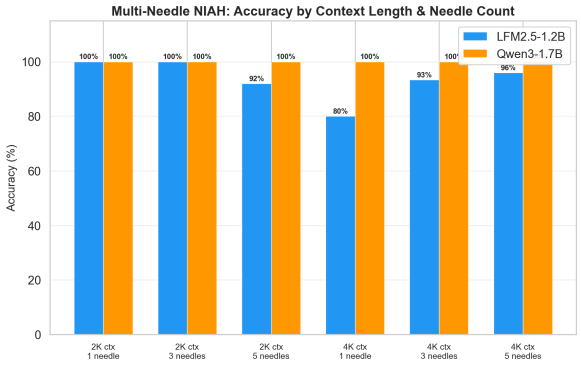
cantly more robust for complex, multi-fact recall.

3.5 Long-Context Perplexity Scaling

Figure 5 shows the long-context perplexity and memory results. A striking divergence emerges beyond 4K tokens: LFM2.5's PPL *degrades* from 16.18 (1K) to 23.50 (8K), while Qwen3 *improves* from 22.11 to 19.31. This pattern supports the hypothesis that convolutions handle local coherence efficiently, but dense attention retains an advantage for capturing longer-range semantic dependencies. This trend is consistent across platforms; supplementary MPS evaluations reproduced the exact same directional behavior, with LFM2.5's perplexity increasing and Qwen3's perplexity decreasing as the context window expanded.



(a) Single-Needle Accuracy Heatmap



(b) Multi-Needle Accuracy

Figure 4: NIAH evaluation across context lengths and needle counts. Both models are strong on single-needle retrieval, while Qwen3 is more reliable on the harder multi-needle setting.

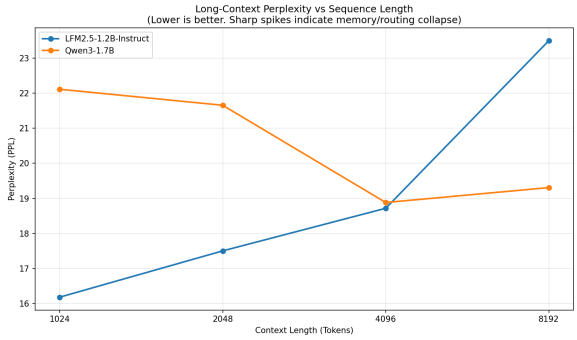
4 Mechanistic Probes and Interpretation

To better understand the architectural drivers behind the empirical efficiency–capability trade-offs, we conducted a series of targeted mechanistic probes. Rather than serving as standalone proofs of architectural laws, these exploratory analyses are designed to interpret the internal behaviors of the models. We focus here on the mechanisms that most directly clarify our main findings: local context sensitivity, KV-cache management, positional dependence, and associative recall. A broader summary of these exploratory probes is provided at the end of this section (Table 9), with full details deferred to Appendix A.

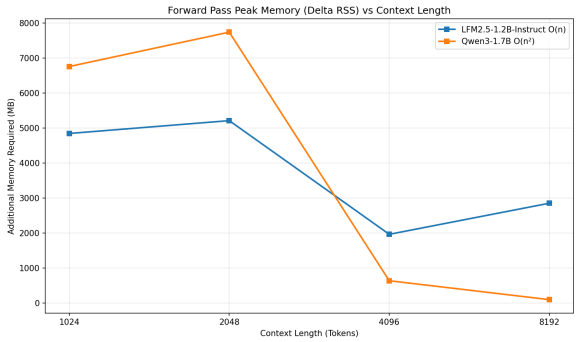
4.1 Longformer-Style Sliding Window Simulation

To test sensitivity to restricted locality, following the sliding-window attention pattern used in Longformer (2), we monkey-patched Qwen3's attention mechanism to mask $Q \cdot K^T$ to a local window of size w.

As shown in Table 6, Qwen3 exhibits a sharp perplexity deterioration at $w<256$. While this serves as a stress test rather than a comparison against a natively trained sliding-window architecture, it



(a) PPL Degradation



(b) Memory Scaling

Figure 5: Long-context perplexity and memory comparison. LFM2.5 retains a memory advantage, but Qwen3 becomes stronger on perplexity as context length increases.

highlights the dense Transformer's heavy reliance on global attention. In contrast, LFM2.5's natively learned convolutions maintain a stable perplexity of 2.21, demonstrating robust local processing without the need for full-sequence attention.

4.2 KV-Cache Efficiency

A direct driver of the observed inference efficiency is KV-cache management. Qwen3 stores KV pairs across all 28 layers, consuming 448 MB at a 2K context length. By design, LFM2.5 caches only in its 6 GQA tracking layers, requiring just 48 MB. This structural **9.3**× reduction demonstrates how hybrid architectures can natively achieve the memory savings typically targeted by post-hoc compression or eviction algorithms in dense models.

4.3 Positional Encoding Ablation (N=100 Texts)

Table 7 and Figure 6 summarize our positional ablation: to assess reliance on explicit positional structures such as rotary positional encoding (RoPE) (3), we ablated all positional information across 100 distinct WikiText-2 (19) segments.

Table 6: Qwen3 PPL under sliding window masks (1024 tokens).

Window w	**Qwen3 PPL**	Δ	**LFM2.5 (Native)**
Full (∞)	1.70	—	2.21
512	1.75	+0.05	
256	1.75	+0.05	
128	68.57	+66.87	
64	159.03	+157.33	
32	**483.18**	+481.48	

Table 7: Positional ablation: PPL (mean ± std, N=100).

Strategy	**Qwen3**	**LFM2.5**
Normal (Ordered)	22.44 ± 8.22	131.58 ± 67.71
All Zeros	$22{,}673 \pm 19{,}027$	472.84 ± 234.44
Fully Shuffled	740.30 ± 274.85	292.31 ± 135.45

Qwen3's perplexity degrades drastically (roughly 1010×) when RoPE is removed, whereas LFM2.5 degrades by only about 3.6×. Furthermore, under fully shuffled positions, LFM2.5 (PPL ≈ 292) performs better than Qwen3 (PPL ≈ 740). This suggests the hybrid model's convolution blocks act as robust local information processors that are significantly less sensitive to structural disruption than dense attention mechanisms.

4.4 Long-Distance Memory Decay (N=100 Facts)

To isolate exact token recall capabilities, we measured associative recall confidence across increasing context distances to evaluate the retention of factual associations (8). Table 8 and Figure 7 summarize the results. Interestingly, **Qwen3 consistently outperforms LFM2.5** on raw associative recall (73–86% vs. 57–63%). Notably, neither model exhibits a clean, distance-dependent decay curve; rather, the bottleneck for LFM2.5 appears to be baseline precision, not long-range memory loss. This directly supports our earlier observation that dense attention maintains an edge in precise, long-range fact retrieval.

4.5 Summary of Mechanistic Findings

Beyond the core probes detailed above, several supplementary experiments yielded interpretive signals consistent with our architectural hypothesis. For instance, frequency-domain analysis indicated a division of labor between higher-frequency convolution blocks and lower-frequency attention layers. Additionally, causal-resilience probing re-

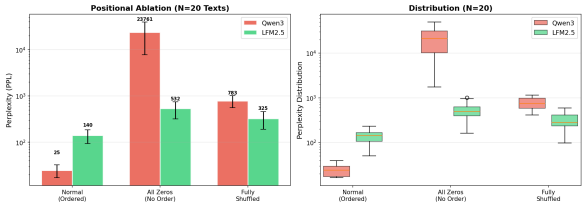


Figure 6: Positional ablation (N=100): bar chart with error bars (left) and box plot (right). The dense model is highly sensitive to positional ablation, while LFM2.5 shows greater structural resilience.

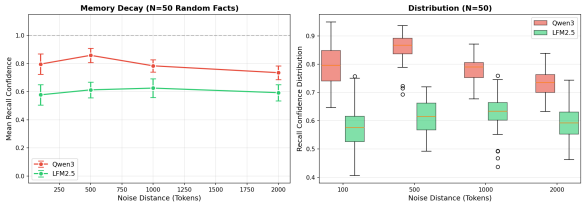


Figure 7: Memory decay (N=100): mean confidence with error bars (left) and box plot (right). Qwen3 demonstrates stronger exact associative recall, with neither model showing a clean monotonic decay pattern over distance.

Table 8: Recall confidence across noise distances (mean $\pm$ std, N=100).

Tokens	Qwen3	LFM2.5
100	$79.8\% \pm 7.2\%$	$60.2\% \pm 8.0\%$
500	$86.9\% \pm 3.6\%$	$61.8\% \pm 7.1\%$
1000	$78.4\% \pm 4.3\%$	$61.8\% \pm 6.8\%$
2000	$73.3\% \pm 4.2\%$	$58.1\% \pm 6.2\%$

vealed that LFM2.5 suffers a much smaller confidence drop under layer ablation compared to Qwen3.

Table 9 summarizes the full suite of our exploratory probes. While sample sizes (N) for some tests are intentionally small by design, collectively they illustrate a consistent pattern: the hybrid model sacrifices some exact recall capabilities in exchange for significant gains in memory efficiency and structural robustness.

5 LoRA Fine-Tuning: Efficiency and Adaptation

We applied LoRA (10) fine-tuning (rank=16, $\alpha = 32$, lr=2×10^{-4}) using an Alpaca-GPT4-style instruction-tuning dataset derived from GPT-4 supervision, following prior instruction-tuning practice (20), on an NVIDIA RTX 4060. The CUDA-side fine-tuning pipeline used an optimized parameter-efficient training implementation.

5.1 Training Resource Comparison

Table 10 summarizes the baseline fine-tuning resource comparison. Under identical baseline settings, training time is effectively matched, but LFM2.5 requires **44% less GPU memory**. This mirrors the inference-side memory savings and establishes a clear training-memory advantage for the hybrid architecture. This efficiency is further supported by normalized training throughput: across our logged runs, LFM2.5 processed approximately

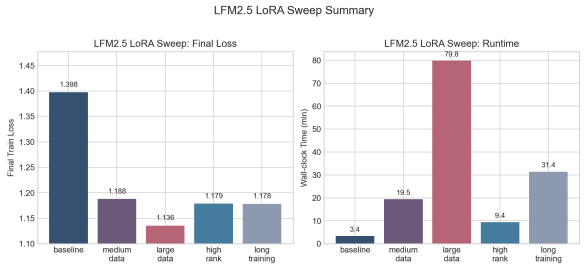


Figure 8: LFM2.5 LoRA sweep summary. Final training loss declines across the tested sweep, while runtime scales predictably with larger data and extended training.

5.95 samples/s compared to 4.96 for Qwen3 in the baseline setup, with the gap widening significantly (6.77 vs. 0.75 samples/s) under higher-budget configurations.

5.2 LFM2.5 Scaling Behavior

Table 11 and Figure 8 detail the internal scaling behavior of LFM2.5 across varying data volumes and training durations. (Note that the end-to-end wall-clock times reported in this sweep include extended logging overhead and are therefore distinct from the strictly paired baseline timings in Table 10). As expected, final training loss declines consistently with increased data and epochs, confirming that the hybrid model responds predictably to standard adaptation scaling.

5.3 Post-Fine-Tuning Quality Evaluation

To assess adaptation fidelity, we evaluated the post-fine-tuning checkpoints using a robust-scored, logit-based evaluation framework (Table 12); supplementary legacy-wrapper results are reported in Table 13. This approach was adopted to eliminate the evaluation brittleness and degenerate artifacts (e.g., zero-score anomalies) occasionally observed in earlier single-token decoding wrappers, ensuring a stable cross-model comparison.

Table 9: Summary of mechanistic experiments.

Experiment	N	**Key Finding**	**Winner**
Sliding Window	1	PPL cliff at $w<256$ in dense model	**LFM2.5**
Frequency Domain	1	Conv=high-freq, Attn=low-freq	Insight
KV-Cache	1	9.3× memory saving	**LFM2.5**
Pos. Ablation	100	1010× vs 3.6× collapse	**LFM2.5**
Attn Sparsity	100	Gini 0.9515 vs 0.8855	**LFM2.5**
Memory Decay	100	80% vs 60% recall precision	Qwen3
Causal Resilience	1000	97.5% vs 12.5% collapse	**LFM2.5**

Table 10: LoRA fine-tuning comparison (200 samples, 2 epochs, rank=16).

Metric	**LFM2.5**	**Qwen3**	**Ratio**
Training Time (s)	63.6	63.7	1.0×
Peak GPU Memory (MB)	3,022	5,364	**LFM2.5** −44%
Final Train Loss	1.430	1.292	Qwen3 lower

Table 11: LFM2.5 LoRA training sweep on RTX 4060.

Config	**Samples**	**Epochs**	**Rank**	**Time (min)**	**Loss**
baseline	200	2	16	3.4	1.398
medium_data	1,000	3	16	19.5	1.188
large_data	5,000	5	16	79.8	1.136
high_rank	1,000	3	64	9.4	1.179
long_training	1,000	10	16	31.4	1.178

Our evaluation highlights a continuation of the baseline efficiency–capability trade-off:

1. **Absolute Post-FT Accuracy:** Qwen3 maintains a commanding lead across the audited multiple-choice tasks. On the `medium_data` checkpoints, Qwen3 achieves 40% MMLU (14), 78% ARC (15), and 49% HellaSwag (16), significantly outperforming LFM2.5 (28%, 30%, and 21%, respectively).
2. **Relative Adaptation Signal:** While LFM2.5 demonstrates modest gains over its original base checkpoint on MMLU and ARC, dense attention appears to provide a more receptive substrate for high-fidelity knowledge adaptation. Supplementary evaluations on math and code generation tasks (GSM8K (17), MBPP (18)), though inherently higher-variance, directionally align with this capability gap.
3. **Post-FT Efficiency:** Despite lower absolute accuracy, LFM2.5 comprehensively preserves its structural advantages post-adaptation. Under a unified CUDA remeasurement, LFM2.5 sustained a throughput of roughly 41.9–44.1 tok/s across the completed sweep, severely outpacing Qwen3's 16.3–16.6 tok/s.

Ultimately, fine-tuning does not alter the fundamental architectural dynamics: the hybrid model remains vastly superior for throughput and deployment cost, while the dense Transformer remains the necessary choice for maximizing absolute output quality.

6 Discussion

6.1 The Efficiency–Capability Trade-Off

Table 14 (Appendix B) provides a compact summary of our primary evidence chain. While aggregated metrics—such as mean multi-needle accuracy and average peak memory—summarize broad benchmark slices rather than a single canonical setting, the overarching narrative is highly consistent: the hybrid architecture trades a degree of exact recall for massive gains in computational efficiency.

Our mechanistic probes suggest this trade-off stems from a division of labor within the hybrid model. Convolution blocks appear to act as robust local information processors, sliding over sequences with reduced reliance on explicit positional machinery. Meanwhile, the sparse GQA layers are reserved for global routing, exhibiting broader attention distributions than their dense counterparts. While serving as an interpretive framework rather than a definitive causal proof, this hypothesis successfully aligns with both the strengths (throughput, memory, robustness) and the weaknesses (multi-fact retrieval, exact associative recall) of the Conv+GQA design.

6.2 Limitations

Several confounding factors and methodological constraints shape the interpretation of these results:

1. **Confounding Variables:** Because we evaluate publicly released checkpoints, LFM2.5 (1.2B) and Qwen3 (1.7B) differ not only in

architecture but also in parameter scale, pretraining corpora, and alignment recipes. Consequently, performance deltas cannot be attributed exclusively to architectural design.

2. **Hardware Scope:** All primary CUDA evaluations were conducted on a single consumer-grade GPU (NVIDIA RTX 4060). Data-center hardware (e.g., A100/H100) with vastly different memory bandwidth profiles may yield different scaling curves and bottleneck behaviors.
3. **Evaluation Variance:** As is common in LLM evaluation, absolute performance figures remain sensitive to the specific prompting wrapper and evaluation framework. While our robust logit-based scoring mitigates the degenerate artifacts seen in early single-token decoding attempts, absolute post-fine-tuning levels—as well as supplementary high-variance tasks like code generation—should be interpreted as methodological sensitivity checks rather than absolute capability ceilings.

6.3 Practical Implications

These findings place the two architectures at distinctly different operating points. The Conv+GQA hybrid is exceptionally attractive for environments where throughput, latency, memory pressure, and adaptation costs are the primary bottlenecks. Conversely, the dense Transformer remains the safer, more capable choice for applications where multi-fact retrieval, exact associative recall, and absolute post-adaptation accuracy are paramount. Rather than declaring a universal winner, this evaluation maps the boundaries of where hybrid models excel and where dense attention remains indispensable.

7 Conclusion

This exploratory study establishes a clear efficiency–capability trade-off between hybrid Conv+GQA and dense Transformer architectures. Under our evaluation protocol, LiquidAI's LFM2.5 defines a highly favorable deployment operating point, delivering up to **11.5**× higher throughput, a **44–47%** reduction in inference memory, a **9.3**× smaller KV cache footprint, and **44%** lower LoRA training memory. Crucially, it achieves these efficiency gains while maintaining flawless single-needle retrieval capabilities across tested contexts.

However, the dense baseline (Qwen3) remains demonstrably more reliable on complex, capability-bound tasks. It consistently outperforms the hybrid model in multi-needle retrieval scenarios, exact associative recall, and robust-scored post-fine-tuning audits. Ultimately, we conclude that the hybrid Conv+GQA architecture represents a powerful, resource-efficient paradigm for standard generative deployment, whereas the traditional dense Transformer remains the necessary architecture for cognitive tasks demanding precise, high-fidelity, long-range reasoning.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. "Attention Is All You Need." *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] I. Beltagy, M. E. Peters, and A. Cohan. "Longformer: The Long-Document Transformer." *arXiv preprint arXiv:2004.05150*, 2020.

[3] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. "RoFormer: Enhanced Transformer with Rotary Position Embedding." *arXiv preprint arXiv:2104.09864*, 2021.

[4] N. Kitaev, Ł. Kaiser, and A. Levskaya. "Reformer: The Efficient Transformer." *International Conference on Learning Representations (ICLR)*, 2020.

[5] M. Beck, K. Pöppel, M. Spanring, A. Auer, O. Prudnikova, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. "xLSTM: Extended Long Short-Term Memory." *arXiv preprint arXiv:2405.04517*, 2024.

[6] A. Behrouz, P. Zhong, and V. Mirrokni. "Titans: Learning to Memorize at Test Time." *arXiv preprint arXiv:2501.00663*, 2025.

[7] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. "Efficient Streaming Language Models with Attention Sinks." *International Conference on Learning Representations (ICLR)*, 2024.

[8] K. Meng, D. Bau, A. Andonian, and Y. Belinkov. "Locating and Editing Factual Associations in GPT." *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[9] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah. "In-Context Learning and Induction Heads." *arXiv preprint arXiv:2209.11895*, 2022.

[10] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations (ICLR)*, 2022.

[11] A. Gu and T. Dao. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces." *arXiv preprint arXiv:2312.00752*, 2023.

[12] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints." *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

[13] A. Yang et al. "Qwen3 Technical Report." *arXiv preprint arXiv:2505.09388*, 2025.

[14] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. "Measuring Massive Multitask Language Understanding." *International Conference on Learning Representations (ICLR)*, 2021.

[15] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. "Think You Have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge." *arXiv preprint arXiv:1803.05457*, 2018.

[16] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. "HellaSwag: Can a Machine Really Finish Your Sentence?" *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, 2019.

[17] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, Ł. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. "Training Verifiers to Solve Math Word Problems." *arXiv preprint arXiv:2110.14168*, 2021.

[18] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, and C. Sutton. "Program Synthesis with Large Language Models." *arXiv preprint arXiv:2108.07732*, 2021.

[19] S. Merity, C. Xiong, J. Bradbury, and R. Socher. "Pointer Sentinel Mixture Models." *arXiv preprint arXiv:1609.07843*, 2016.

[20] B. Peng et al. "Instruction Tuning with GPT-4." *arXiv preprint arXiv:2304.03277*, 2023.

Appendix

A Extended Mechanistic Interpretability Probes

This section collects several supplementary probes that provide additional descriptive evidence about how the two models organize context processing internally. Unlike the main mechanistic probes in the body of the paper, these analyses are included primarily for interpretive context rather than as part of the core evidence chain.

A.1 Attention Mass Decay and Induction Heads

Figure 9 summarizes the attention-mass and induction-head analyses. We first examine how attention mass varies as a function of token distance. Tracking attention weight $\rho(d)$ over sequences of length 2048 suggests that LFM2.5 suppresses very local attention more strongly, consistent with the idea that short-range processing is partly delegated to convolutional layers, while its remaining attention layers participate more selectively in longer-range routing.

We also probed for induction-head-like behavior in both models (9). Qwen3 shows a strong induction head at Layer 16, Head 14 (score: 0.997), while LFM2.5 shows a similarly strong pattern at Layer 2, Head 1 (score: 0.963), despite using substantially fewer attention layers overall. This suggests that induction-style pattern copying can still emerge clearly within the hybrid architecture.

A.2 Gradient-Based Effective Receptive Field

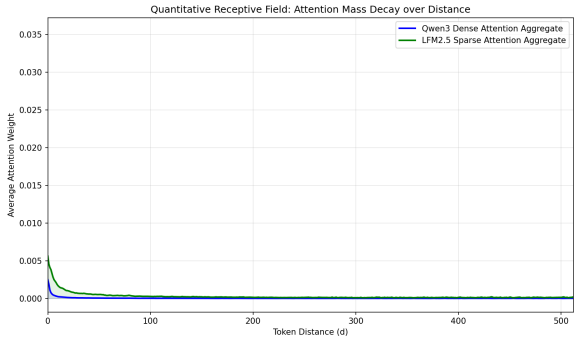
We next compute a continuous gradient-based effective receptive field (ERF),

$$\mathrm{ERF}(i) = \left\| \frac{\partial \mathbf{y}_{\text{last}}}{\partial \mathbf{x}_i} \right\|_2 .$$

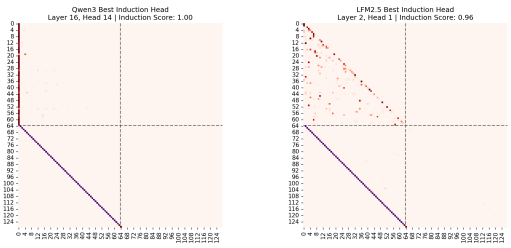
Figure 10 shows that, in this visualization, LFM2.5 exhibits a more attenuated and structured ERF profile than Qwen3. This pattern is consistent with the interpretation that convolution-heavy processing reduces some forms of continuous long-range interaction, even when the model retains limited attention-based global routing.

A.3 Internal Representation Geometry

Figure 11 shows that, to compare representation organization across layers, we computed the pairwise cosine similarity matrix over hidden states from all



(a) Attention mass decay $\rho(d)$



(b) Induction head discovery

Figure 9: Attention-distance structure and induction-head analysis.

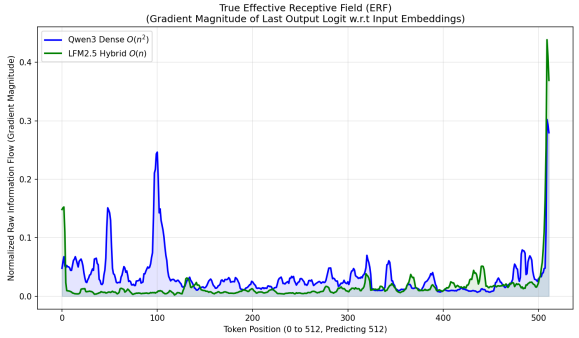
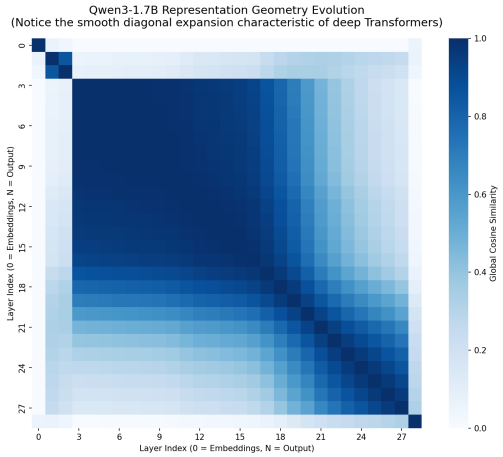


Figure 10: Gradient-based effective receptive field (ERF).

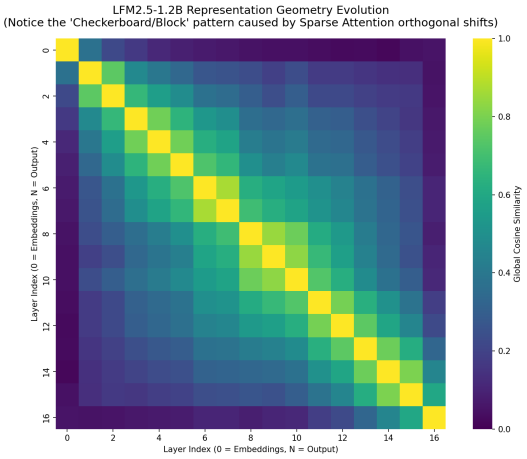
layers. Qwen3 shows a smoother diagonal structure, consistent with more gradual residual refinement across depth. LFM2.5, by contrast, exhibits a more block-like pattern, with sharper transitions around GQA layer boundaries. This difference is descriptive rather than causal, but it is consistent with the broader picture of a more modular internal organization in the hybrid model.

A.4 SVD Spectral Entropy

Figure 12 shows that the layer-wise spectral-entropy curves differ across the two models, indicating distinct representational trajectories over depth. However, the pattern is less clean than a simple alternating low-high interpretation. We therefore treat this figure as descriptive evidence about representational organization rather than as strong confirmation of a single mechanistic hypothesis.



(a) Qwen3 (dense)



(b) LFM2.5 (hybrid)

Figure 11: Layer-wise representation geometry measured by pairwise cosine similarity.

A.5 Autoregressive Latency and Attention Sinks

Figure 13 shows that a token-level latency profile further illustrates the deployment-side difference between the two models. In this measurement setup, LFM2.5 maintains a flatter per-token latency curve, whereas Qwen3 slows more noticeably as context grows and KV-cache costs accumulate. A later Apple `MPS` decode-latency spot check using a lightweight manual unroll (`2` trials, `48` decoded tokens after warmup) was qualitatively consistent with the same pattern: both models had similar mean decode latency over the completed curve (`135.9` ms for LFM2.5 vs. `135.2` ms for Qwen3), but Qwen3 ended with a substantially slower tail

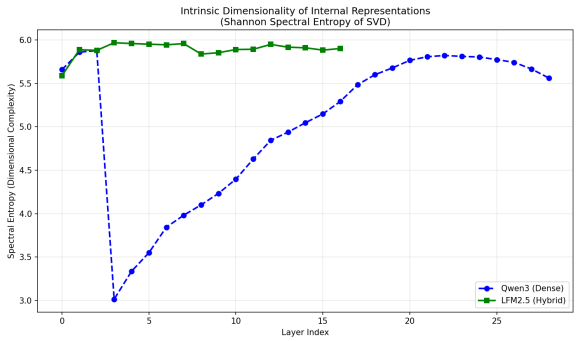


Figure 12: SVD spectral entropy by layer. The two models show different layer-wise entropy trajectories, but the figure is best interpreted descriptively rather than as a clean confirmation of a single alternating-pattern hypothesis.

Table 12: Robust-scored CUDA-side cross-model post-fine-tuning comparison (100-example audit).

Model	Config	MMLU	ARC	HellaSwag
Qwen3	original	42%	79%	45%
Qwen3	baseline	38%	81%	44%
Qwen3	medium_data	40%	78%	49%
LFM2.5	original	25%	20%	26%
LFM2.5	baseline	27%	27%	24%
LFM2.5	medium_data	28%	30%	21%

token (`232.3` ms vs. `150.2` ms), whereas LFM2.5 remained flatter near the end of the sequence.

We also examined attention sinks (7). Both models exhibit strong initial-token attention concentration, indicating that Softmax-based attention remains a structural bottleneck even in the hybrid setting.

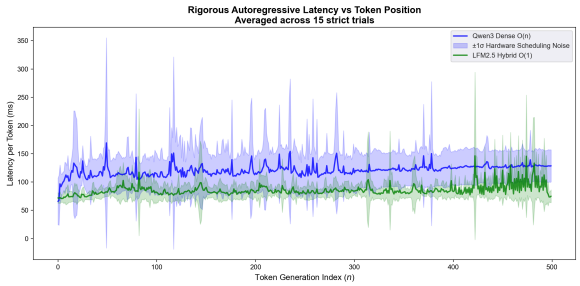
B Supplementary Post-Fine-Tuning Tables

Appendix note. These tables are retained as supplementary reference material. The manuscript's main cross-model post-fine-tuning comparison relies on the robust-scored audit in Table 12, while the legacy-wrapper sweep and compact scorecard are kept here to reduce layout pressure in the main text.

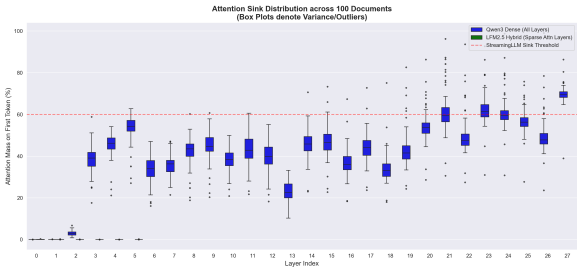
C Implementation Details and Reproducibility Notes

C.1 Default Settings for Central Runs

For the main reported settings, the script-level defaults were as follows. The baseline CUDA LoRA comparison used an Alpaca-GPT4-style



(a) Autoregressive latency



(b) Attention sink distribution

Figure 13: Token-level latency profiling and attention-sink analysis.

Table 13: Legacy-wrapper post-fine-tuning benchmark evaluation (LFM2.5 sweep).

Config	MMLU	ARC	Throughput (tok/s)
Original (no FT)	22%	64%	46.8
baseline (200)	36%	64%	50.7
medium_data (1K)	**38%**	68%	49.7
large_data (5K)	32%	**70%**	6.7
high_rank (r=64)	**38%**	68%	6.5
long_training (10ep)	34%	68%	6.8

instruction-tuning dataset derived from GPT-4 supervision (20), LoRA rank 16, $\alpha = 16$ or 32 depending on the script family, learning rate 2×10^{-4}, per-device batch size 2, gradient accumulation 4, warmup of 10 steps, maximum sequence length 1024, and seed / random state `3407`.

The main CUDA inference benchmark used greedy decoding with output length 256, `3` warmup runs, and `10` timed runs per configuration. TTFT was measured separately using a one-token greedy generation call, and peak CUDA memory was read from `torch.cuda.max_memory_allocated()`.

The later fixed-token Apple `MPS` rerun used batch size 1, exactly 64 generated tokens, `1` warmup run, and `3` timed runs, with peak host memory approximated by process RSS.

C.2 Evaluation Wrappers

The repository contains more than one benchmark wrapper family, which is relevant for in-

Table 14: Summary scorecard across the main evaluation dimensions.

Dimension	LFM2.5	Qwen3	Winner
Single NIAH	100%	100%	Tie
Multi-Needle NIAH (mean)	93.5%	100%	Qwen3
Throughput (bs=4, 4K)	72.7 tok/s	6.3 tok/s	**LFM2.5 (11.5×)**
Avg Peak Memory (bs=1, 128–4096)	2,403 MB	4,354 MB	**LFM2.5 (−45%)**
KV-Cache (2K)	48 MB	448 MB	**LFM2.5 (9.3×)**
FT Memory (LoRA)	3,022 MB	5,364 MB	**LFM2.5 (−44%)**
Post-FT MMLU (robust audit, medium_data)	28%	40%	Qwen3
Post-FT Throughput (unified CUDA rerun, best completed ckpt.)	44.1 tok/s	16.6 tok/s	**LFM2.5**
Associative Recall (N=100)	60%	80%	Qwen3

terpreting the post-fine-tuning results. The earlier multiple-choice evaluation path truncated prompts to 512 tokens, used greedy decoding with `max_new_tokens=1`, and scored only the first decoded character against the expected option label. Later cleanup-stage reruns instead used a logit-based option-scoring path for Apple `MPS` sanity checks and CUDA-side audits on merged checkpoints.

Accordingly, the robust-scored CUDA audit is treated in the main text as the canonical cross-model post-fine-tuning quality comparison, while the earlier one-token wrapper is retained only as a diagnostic artifact.

C.3 Scope of Interpretation

These implementation details improve reproducibility, but they do not eliminate all wrapper and pipeline mismatches discussed in the main text. The paper should therefore still be read as an exploratory released-checkpoint comparison under a shared evaluation protocol rather than as a fully unified architecture-only benchmark suite.

D Use of AI-Based Tools

- **DeepSeek**: DeepSeek models were consulted for guidance on more formal ways of organizing the conclusion section. Any suggested structures were adapted and rewritten entirely in my own words.

- **QuillBot**: QuillBot was used sparingly to rephrase selected sentences for readability and flow. All suggestions were manually reviewed and revised to ensure consistency with my intended meaning and academic writing style.

- **DeepL and Youdao Translation**: DeepL and Youdao Translation were used to translate a small number of technical terms and short phrases from Chinese into English during drafting. These translations were checked and incorporated only after manual verification.