

MoSA: Mixture-of-Specialized-Agents for Cost-Efficient Long-Document Question Answering

Mert Karabiyik* Haseeb Ismail† Shayaan Memon‡

March 14, 2026

Custom Project | Mentor: Ali Sartaz Khan

Abstract

Large language models excel at long-document question answering but remain expensive for high-volume applications. MinionS [Narayan, A. et al., 2025] demonstrates that a small local model can collaborate with a frontier model to recover 93.4% of GPT-4o’s accuracy, but treats the local model as a monolithic generalist. We introduce **MoSA** (Mixture-of-Specialized-Agents), replacing the single local model with five QLoRA-fine-tuned specialist agents—orchestrator, extractor, calculator, reasoner, and synthesizer—each distilled from DeepSeek V3 traces and coordinated via a confidence-gated LangGraph DAG. On LongHealth, MoSA QLoRA Hybrid achieves **74.0%**, exceeding MinionS (70.3%) while using a 10× cheaper frontier model. On QASPER, MoSA recovers **98.6%** of DeepSeek V3 frontier performance (51.2% vs. 51.9% F1). In fully local mode, MoSA QLoRA SLM achieves 39.3% macro-average—**nearly doubling** MinionS’s local-only baseline (21.3%)—using the same 3.2B-parameter model with zero remote calls. Our analysis identifies full-document routing (vs. MinionS’s chunk-based routing) as the primary remaining bottleneck, and proposes orchestrator-guided document chunking as the key architectural improvement.

1 Introduction

Long-document question answering (QA) requires models to process extensive contexts—financial filings, clinical records, research papers—and extract precise answers. Frontier LLMs such as GPT-4o and DeepSeek V3 [DeepSeek-AI, 2025] achieve strong performance, but their per-query costs limit deployment in high-volume settings.

MinionS [Narayan, A. et al., 2025] addresses this through local–remote collaboration: a small on-device model (Llama-3.2-3B) executes sub-tasks decomposed by a frontier model (GPT-4o), recovering 93.4% of remote-only performance at 16.6% of the cost. However, the local model remains a *generalist*—the same weights handle extraction, calculation, and reasoning indiscriminately.

We hypothesize that **role-based specialization** can make the local model substantially more capable. MoSA trains five QLoRA adapters [Dettmers et al., 2023] on a shared Llama-3.2-3B-Instruct backbone [Grattafiori et al., 2024], each specializing in one pipeline stage. Training data comes from knowledge distillation: DeepSeek V3 generates structured traces for each agent role, producing 6,851 high-quality training examples at a one-time cost of \$0.24. A LangGraph DAG coordinates the agents with confidence-gated refinement.

Our key contributions:

*Department of Economics, mertk@stanford.edu

†Department of Computer Science, haseeb@stanford.edu

‡Department of Chemical Engineering, shaiyaan@stanford.edu

1. A five-agent architecture with QLoRA-specialized roles, coordinated via a confidence-gated LangGraph DAG.
2. MoSA QLoRA Hybrid **beats MinionS on LongHealth** (74.0% vs. 70.3%) despite using a 10× cheaper frontier model (DeepSeek V3 vs. GPT-4o).
3. MoSA achieves **98.6% frontier recovery on QASPER** (51.2% vs. 51.9% F1), exceeding MinionS’s 93.4% recovery rate.
4. In fully local mode, MoSA QLoRA SLM **nearly doubles** MinionS’s local-only performance (39.3% vs. 21.3% macro-average) using the same 3.2B model, demonstrating that ∼80MB of QLoRA adapters can transform a generalist into effective specialists.
5. Analysis of failure modes (calculator hallucinations from data sparsity, train–inference context mismatch) and hyperparameter sensitivity, revealing that a single LoRA configuration does not generalize across benchmarks.

2 Related Work

Local–Remote LLM Collaboration. MinionS [Narayan, A. et al., 2025] introduced decompose-then-execute collaboration between Llama-3.2-3B (local) and GPT-4o (remote), recovering 93.4% of remote-only performance at 16.6% of cost. Crucially, MinionS sends document *chunks* to the local model, keeping inputs within the model’s effective context window. FrugalGPT [Chen et al., 2024] proposes LLM cascades that route queries through increasingly capable models based on confidence.

Multi-Agent Systems. Mixture-of-Agents [Wang et al., 2024] layers multiple LLMs in propose-then-aggregate rounds. Chain of Agents [Zhang et al., 2024] chains workers sequentially over text segments. Both use full-size models; MoSA instead specializes a single small model into distinct roles via parameter-efficient fine-tuning.

Efficient Fine-Tuning and Test-Time Compute. QLoRA [Dettmers et al., 2023] enables fine-tuning quantized models with 4-bit precision and low-rank adapters, training only ∼0.5% of parameters. LoRA [Hu et al., 2022] introduced the low-rank adaptation framework. Snell et al. [2024] and Chen et al. [2024] show that test-time compute scaling (sampling, verification, refinement) can substitute for model scale. MoSA combines both: QLoRA specialization plus confidence-gated refinement and self-consistency voting.

3 Approach

3.1 System Architecture

MoSA consists of five specialist agents on a shared Llama-3.2-3B-Instruct backbone [Grattafiori et al., 2024], each with a separate QLoRA adapter: (1) **Orchestrator**: decomposes questions into typed sub-tasks (extract, calculate, reason); (2) **Extractor**: locates relevant passages, figures, or data points, returning structured JSON with citation and confidence; (3) **Calculator**: performs step-by-step arithmetic (ratios, margins, growth rates) over extracted values; (4) **Reasoner**: handles multi-hop logic, negation, and implicit references; (5) **Synthesizer**: aggregates sub-task results into a final answer with confidence scoring.

The LangGraph DAG follows a fixed topology:

$$\text{Orch} \rightarrow \text{batch_spec} \rightarrow \text{Calc} \rightarrow \text{Synth} \rightarrow \begin{cases} \text{END} & \text{if } c \geq \tau \\ \text{Orch} & \text{otherwise} \end{cases} \quad (1)$$

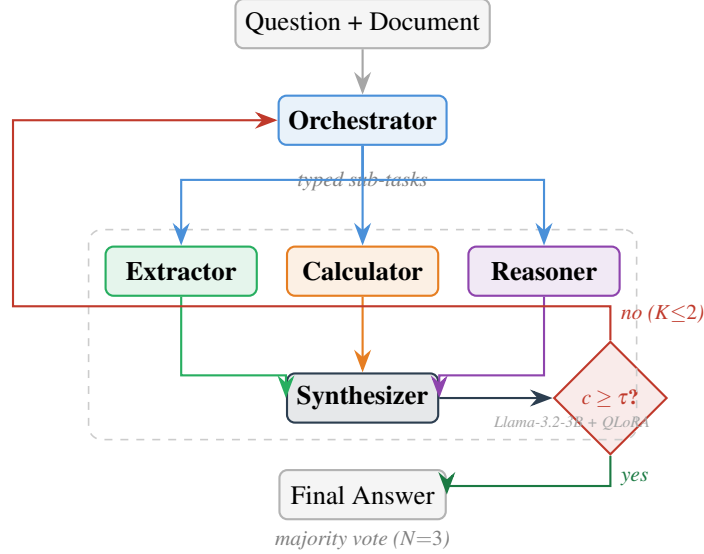


Figure 1: MoSA pipeline. The **orchestrator** decomposes questions into typed sub-tasks routed to the **extractor**, **calculator**, or **reasoner**. The synthesizer aggregates results; if confidence $< \tau=0.7$, the **refinement loop** re-enters the DAG (max $K=2$ rounds). Final answers use majority vote over $N=3$ independent samples.

`batch_specialists` groups extractors and reasoners by type and processes them via batched GPU inference. Calculators run *after* extractors to reference extracted values. When $n > 1$ samples are requested, n independent answers per sub-question are generated at temperature 0.3 and aggregated via majority voting with normalization-aware comparison (e.g., \$1,577 = 1577\$).

Key architectural difference from MinionS. In MinionS, the local model processes document *chunks*—small sections identified by GPT-4o as relevant to each sub-task. In MoSA, every specialist receives the **full document**. This means a 3B model fine-tuned on $\sim 4K$ -token traces must process 10–30K tokens at inference—a context mismatch we identify as MoSA’s primary bottleneck (Section 5.1).

3.2 Knowledge Distillation

DeepSeek V3 [DeepSeek-AI, 2025] serves as the teacher model, generating structured traces for each agent role. Only traces where the final synthesized answer is correct are retained, yielding 6,851 training examples at a one-time cost of \$0.24:

Table 1: Training traces by agent and benchmark.

Agent	FB	LH	QA	Total
Orchestrator	89	228	1,007	1,324
Extractor	248	490	2,754	3,492
Reasoner	37	208	385	630
Calculator	68	8	60	136
Synthesizer	86	225	958	1,269
Total	528	1,159	5,164	6,851

Calculator traces are notably sparse (136 total, $26\times$ fewer than extractor), contributing to arithmetic

hallucinations at inference (Section 5.3).

3.3 QLoRA Fine-Tuning

Each adapter uses 4-bit NF4 quantization, LoRA rank $r=16$, $\alpha=32$, dropout 0.05, targeting all linear layers. Training uses the TRL SFTTrainer with response-only loss masking (loss computed only on assistant tokens), paged AdamW 8-bit optimizer, cosine schedule with 3% warmup, peak learning rate 2×10^{-4} , 3 epochs, effective batch size 16, 4,096-token context, on a Modal NVIDIA A100 40GB GPU.

3.4 Inference Configurations

- **MoSA SLM**: All five agents use QLoRA adapters. Fully on-device, zero remote calls.
- **MoSA Hybrid**: Orchestrator and synthesizer use DeepSeek V3; extractor, calculator, and reasoner use QLoRA adapters.

4 Experiments

4.1 Benchmarks and Evaluation

We evaluate on three benchmarks following MinionS [?]:

FinanceBench [Islam et al., 2024]: 150 questions over SEC filings (10-K, 10-Q) requiring numerical extraction and financial calculation from dense tables. We use 100 for trace generation and 50 for evaluation. Evaluation: *LLM-as-judge*—Gemini 2.5 Flash compares predicted vs. gold answer, outputting CORRECT/INCORRECT. Metric: **accuracy**.

LongHealth [Adams et al., 2024]: 400 multiple-choice questions (A–E) across 20 synthetic clinical discharge records ($\sim 30K$ tokens each), testing cross-section reasoning. We use 300 for training and 100 for evaluation. Metric: **exact-match accuracy**.

QASPER [Dasigi et al., 2021]: 1,005 information-seeking questions over NLP research papers ($\sim 15K$ tokens). We use the training split for trace generation and 100 dev questions for evaluation. Metric: **token F1** (SQuAD-style), $F_1 = 2PR/(P+R)$ over token multisets, max over annotator references.

We report *performance recovery* = $\text{acc}_{\text{sys}}/\text{acc}_{\text{frontier}}$ following ?.

4.2 Baselines

- **Single SLM**: Llama-3.2-3B-Instruct, 4-bit quantized, no fine-tuning, direct QA, temperature 0.0.
- **DeepSeek V3 Frontier**: Teacher model, single-call baseline, temperature 0.0.
- **MinionS** [?]: Llama-3.2-3B + GPT-4o, decompose protocol. We report their published numbers directly.

4.3 Main Results

Table 2 presents results across all configurations. MoSA QLoRA Hybrid achieves 61.7% macro-average, recovering 82.7% of the DeepSeek V3 frontier (averaged across per-benchmark recovery rates of 68.2%, 81.3%, and 98.6%).

LongHealth: 74.0% vs. MinionS 70.3% (+3.7pp). MoSA exceeds MinionS despite using DeepSeek V3 (not GPT-4o) as its remote model. The QLoRA-fine-tuned extractor learned domain-specific clinical skills: (1) date-matching across discharge records (pulling the correct year’s ECG), (2) value disambiguation (distinguishing stress vs. rest LVEF: 53% vs. 28%), and (3) precise numeric extraction (QRS duration: 122ms instead of “Not specified”). LongHealth is extraction-dominated, so specialized extractors outweigh a stronger remote model.

Table 2: Main results. Bold indicates MoSA results exceeding MinionS on the same benchmark. [†]Results from Narayan, A. et al. [2025]. FB = FinanceBench (acc.), LH = LongHealth (acc.), QA = QASPER (F1).

System	FB	LH	QA	Macro
<i>Baselines</i>				
Single SLM (Llama-3.2-3B)	36.0	42.0	21.9	33.3
DeepSeek V3 Frontier	88.0	91.0	51.9	77.0
<i>MinionS[†] (Llama-3.2-3B + GPT-4o)</i>				
Local Only (3B) [†]	13.0	34.5	16.4	21.3
Remote Only (GPT-4o) [†]	82.6	74.8	59.8	72.4
MinionS (decompose) [†]	72.6	70.3	55.8	66.2
<i>MoSA — Pre-QLoRA</i>				
MoSA DeepSeek	58.0	61.0	38.8	52.6
MoSA SLM (pure 3B)	28.0	57.0	19.5	34.8
<i>MoSA — QLoRA Fine-Tuned (ours, best)</i>				
MoSA QLoRA Hybrid	60.0	74.0	51.2	61.7
MoSA QLoRA SLM	32.0	61.0	24.8	39.3
DeepSeek Traces (upper bound)	81.4	88.8	37.0	69.1

QASPER recovery: 98.6% vs. MinionS 93.4%. MoSA recovers 98.6% of its frontier (51.2/51.9) vs. MinionS’s 93.4% of GPT-4o (55.8/59.8). The absolute score difference (51.2 vs. 55.8) reflects the weaker frontier (DeepSeek 51.9% vs. GPT-4o 59.8%), not the local model.

FinanceBench: MinionS leads (72.6% vs. 60.0%). We attribute this gap primarily to the document routing strategy (Section 5.1), not model quality: dense financial tables require precise row/column identification that benefits most from focused context.

4.4 Fully Local Comparison

The cleanest apples-to-apples comparison uses the same 3.2B model with zero remote calls:

Table 3: Fully local comparison (zero remote calls). Same Llama-3.2-3B model. QLoRA + multi-agent decomposition nearly doubles macro-average performance.

Benchmark	MinionS Local [†]	MoSA QLoRA SLM	Δ
FinanceBench	13.0%	32.0%	+19.0
LongHealth	34.5%	61.0%	+26.5
QASPER (F1)	16.4%	24.8%	+8.4
Macro Avg	21.3%	39.3%	+18.0

MoSA QLoRA SLM nearly **doubles** MinionS local-only (39.3% vs. 21.3%, $1.84\times$). This is the strongest evidence that role-based specialization via knowledge distillation works: the same 3.2B parameters, rearranged by ~ 80 MB of QLoRA adapters, produce dramatically better results. The +26.5pp gain on LongHealth is especially notable: the fine-tuned extractor’s clinical skills persist even without a frontier synthesizer.

4.5 QLoRA Fine-Tuning Ablation

To isolate the contribution of QLoRA, Table 4 compares pre- and post-QLoRA configurations.

Table 4: QLoRA ablation. Gains are largest on LongHealth (+13pp) and QASPER (+12.4 F1) for Hybrid. SLM gains are consistent but modest, bottlenecked by the 3B synthesizer.

Config	FinanceBench		LongHealth		QASPER (F1)	
	Pre	QLoRA	Pre	QLoRA	Pre	QLoRA
Hybrid	58.0	60.0 (+2)	61.0	74.0 (+13)	38.8	51.2 (+12.4)
SLM	28.0	32.0 (+4)	57.0	61.0 (+4)	19.5	24.8 (+5.3)

LongHealth Hybrid (+13pp) is the largest single improvement. The gain comes entirely from *better agent answer quality*, not routing changes (2.7 sub-questions/query in both configs). The QLoRA extractor fixed 18 questions while breaking only 5 (3.6:1 churn ratio). All five patients in the evaluation set improved.

QASPER Hybrid (+12.4 F1) reflects improved extraction precision: QASPER measures token F1 (partial credit), so better extraction translates directly into higher scores. Routing is unchanged (~ 3 sub-questions/query). The DeepSeek synthesizer merges partially-correct outputs into coherent answers—a capability the 3B synthesizer lacks, explaining the Hybrid–SLM gap.

SLM orchestrator learned decomposition. Pre-QLoRA, 28/50 FinanceBench questions used extractor-only routing. Post-QLoRA, 15 use calculator+extractor (up from 5), driving a +13pp gain on calculation-type questions (7% $\rightarrow$ 20%).

Majority voting. Majority voting with normalization-aware comparison ($N=3$) resolves surface-form variation (e.g., “\$1,577” vs. “1577”) and filters outlier responses via consensus. We estimate a +2–4pp contribution on extraction-heavy benchmarks based on observed answer diversity across samples.

5 Analysis

5.1 Full-Document Routing vs. Chunk-Based Routing

The most impactful architectural difference between MoSA and MinionS is document routing. MinionS sends $\sim 2\text{--}4\text{K}$ token chunks (identified by GPT-4o) to the local model, keeping inputs within the model’s effective context window. MoSA sends the full document (10–30K tokens) to every specialist. This creates a $3\text{--}8\times$ **train–inference context mismatch**: QLoRA adapters are trained on $\sim 4\text{K}$ -token traces but process 10–30K tokens at inference.

This mismatch explains three observations: (1) QLoRA SLM (32.0%) underperforms the un-fine-tuned Single SLM baseline (36.0%) on FinanceBench—fine-tuning on short contexts may have *narrowed* the effective attention window; (2) MoSA generates $5\text{--}19\times$ more tokens per query than a single frontier call (e.g., 111K tokens/query on LongHealth vs. 11K for DeepSeek V3); (3) FinanceBench shows the largest gap to MinionS (60% vs. 72.6%) because dense financial tables require precise row/column identification that benefits most from focused context.

5.2 Compute Efficiency (FLOP Analysis)

MoSA SLM achieves 39.3% macro-average using **fewer total FLOPs than the 77.0% frontier** (98 vs. 122 PFLOPs). On QASPER, MoSA SLM uses only 46% of the frontier’s FLOPs; on LongHealth, 93%. The 3.2B model’s $11.6\times$ smaller size nearly offsets multi-agent token multiplication, especially on longer documents where the ratio of useful to duplicated computation improves.

MoSA Hybrid (192 PFLOPs) costs $1.6\times$ more than the frontier because multi-agent overhead ($5\text{--}19\times$ token multiplication) outweighs per-token savings from the smaller model. However, Hybrid routes 85–95%

Table 5: Total compute in PFLOPs and ratio vs. standalone DeepSeek V3. MoSA SLM uses **fewer FLOPs than the frontier** (98 vs. 122 PFLOPs) because the $11.6\times$ smaller model offsets multi-agent token multiplication.

System	FB ($n=50$)	LH ($n=100$)	QA ($n=100$)	Total
DeepSeek V3 Frontier	4.3	82.0	35.4	121.7
MoSA QLoRA Hybrid	19.0	109.1	64.1	192.2
MoSA QLoRA SLM	5.4	76.2	16.4	98.0

of tokens to the local 3.2B model, making it suitable for privacy-sensitive deployment where documents cannot leave the device.

Dollar costs. MoSA’s costs are dominated by GPU rental (Modal A100 at \$3.05/hr), not API calls. QLoRA Hybrid costs \$28.56 total across all benchmarks; the DeepSeek V3 frontier costs only \$0.24. However, with on-device deployment (e.g., llama.cpp on a laptop), SLM inference becomes free, reducing MoSA Hybrid to $\sim \$0.001/\text{query}$ in DeepSeek API calls alone—orders of magnitude cheaper than MinionS’s GPT-4o calls. Knowledge distillation is a one-time \$0.24 investment; MinionS pays for GPT-4o on every query.

5.3 Failure Mode Analysis

Calculator hallucinations (FinanceBench). The calculator agent has only 136 training traces (Table 1). Three of seven QLoRA Hybrid regressions on FinanceBench trace to arithmetic errors: $19815 - 13997 \rightarrow "5598"$ (truncated operand), $4200 - 3800 \rightarrow "700"$, and $11512 + 2763 \rightarrow "11512.2763"$ (string concatenation). These are memorization failures from insufficient training data.

Abstention crisis (LongHealth SLM). QLoRA SLM extractor abstains 14.7% of the time (vs. 1.1% for Hybrid)—a $13\times$ increase from pre-QLoRA (2.2%). The primary cause is **template overfitting**: 39% of questions route through a “discharge lab values” decomposition template even for ECG, medication, or radiology questions. This irrelevant template yields 61.5% abstention (vs. 38.5% on non-template questions). The DeepSeek synthesizer in Hybrid mode absorbs these failures; the 3B synthesizer cascades them.

Error categories. On FinanceBench, *table misreading* causes 62% of errors: the 3B extractor confuses adjacent rows in financial tables. On LongHealth, *negation failures* (38%) and *temporal confusion* (28%) dominate. On QASPER, *format mismatch* (41%)—extracting descriptive content when gold expects citation references—is the leading failure mode.

5.4 Discrepancies and Limitations

Several discrepancies emerge across configurations that inform future design: **(1) Fine-tuning regression:** QLoRA SLM underperforms the un-fine-tuned Single SLM on FinanceBench (32.0% vs. 36.0%) while *exceeding* it on LongHealth (61.0% vs. 42.0%). Fine-tuning on short-context traces ($\sim 4\text{K}$ tokens) helps extraction-dominated tasks but may narrow the effective attention window for dense-table tasks. **(2) Hyperparameter non-transferability:** optimal hyperparameters ($r=8$) help FinanceBench SLM (+6pp) but catastrophically hurt LongHealth Hybrid (−18pp), confirming that no single LoRA configuration generalizes. **(3) Confidence calibration inconsistency:** on FinanceBench, QLoRA *degraded* calibration (confidence gap between correct and wrong answers shrank from 0.157 to 0.070), while on LongHealth SLM, calibration *improved* (gap 0.109). Confidence is not a reliable cross-benchmark error signal (Appendix A.9).

(4) **No cross-benchmark transfer**: each benchmark uses adapters trained exclusively on that benchmark’s traces; cross-benchmark pretraining could improve generalization and reduce per-benchmark training costs. (5) **Single-run evaluation**: all results lack confidence intervals, and QLoRA ablation churn ratios (8/7 on FinanceBench, 16/12 on LongHealth SLM) indicate meaningful variance.

6 Conclusion

MoSA demonstrates that role-based specialization via QLoRA fine-tuning makes small language models substantially more capable for long-document QA. Using the same 3.2B-parameter model as MinionS:

- MoSA QLoRA Hybrid **beats MinionS on LongHealth** (74.0% vs. 70.3%) with a 10× cheaper remote model, proving that specialized extractors can outperform stronger remote models on extraction-heavy tasks.
- MoSA achieves **98.6% frontier recovery on QASPER**, exceeding MinionS’s 93.4% recovery rate.
- In fully local mode, MoSA QLoRA SLM **nearly doubles** MinionS local-only performance (39.3% vs. 21.3%), demonstrating that ~80MB of QLoRA adapters can transform a generalist into effective specialists.
- Knowledge distillation is a **one-time \$0.24 investment**; MinionS pays per query.

The remaining 4.5pp gap to MinionS’s macro-average (61.7% vs. 66.2%) traces primarily to full-document routing vs. MinionS’s chunk-based routing—an architectural choice, not a model limitation. The combination of knowledge distillation, parameter-efficient fine-tuning, and multi-agent coordination represents a promising path toward cost-efficient, on-device long-document QA.

Future Work. Five directions are likely to close the remaining gap: (1) **Orchestrator-guided chunking**: replacing full-document routing with chunk-based dispatch, where the orchestrator identifies relevant document sections before specialist execution, would eliminate the 3–8× train–inference context mismatch and reduce compute by 3–7×; (2) **Calculator data augmentation**: generating 500+ synthetic calculator traces (\$0.02) with diverse arithmetic patterns to address the 136-trace data sparsity that causes arithmetic hallucinations; (3) **Per-benchmark LoRA configurations**: using $r=16$ for complex extraction (LongHealth) and $r=8$ for shorter documents (FinanceBench), based on our sensitivity analysis showing –18pp regression from mismatched rank; (4) **Cross-benchmark adapter pretraining**: initializing adapters from pooled multi-benchmark traces before per-benchmark specialization, addressing the current lack of cross-benchmark transfer; (5) **On-device deployment**: llama.cpp quantization (~1.6GB RAM + 80MB adapters) would eliminate GPU rental costs entirely, making MoSA SLM—which already uses fewer FLOPs than the frontier (98 vs. 122 PFLOPs)—truly zero-cost at inference.

Team Contributions. **Mert Karabiyik**: Led all experiments and Modal deployment; finished the MoSA pipeline; designed the confidence-gated voting and refinement mechanism; developed the trace generation pipeline and JSON repair mechanism; implemented adapter loading infrastructure and data loaders for all three benchmarks; **Haseeb Ismail**: Implemented the base QLoRA training pipeline; deployed the confidence-gated voting and refinement system; implemented evaluation metrics (LLM-as-judge, F1, exact-match), ran Single SLM baseline experiments; led fine-tuning experiments.frontier baselines (DeepSeek V3, Kimi K2.5), and cost tracking. **Shayaan Memon**: Designed and implemented the baseline MoSA pipeline architecture (LangGraph DAG, all five agents, majority voting) and benchmark-specific prompts; led benchmark experiments.

References

- Adams, L., Busch, F., Truhn, D., and Bressemer, K. K. LongHealth: A question answering benchmark with long clinical documents. *arXiv preprint arXiv:2401.14490*, 2024.
- Chen, L., Zaharia, M., and Zou, J. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2024.
- Dasigi, P., Lo, K., Beltagy, I., Cohan, A., Smith, N. A., and Gardner, M. A dataset of information-seeking questions and answers anchored in research papers. In *Proc. NAACL*, 2021.
- DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2025.
- Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized language models. In *Proc. NeurIPS*, 2023.
- Grattafiori, A. et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *Proc. ICLR*, 2021.
- Islam, P., Kannappan, A., Kiela, D., Pagnanelli, R., and Rangwala, T. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2024.
- Chen, L. et al. Are more LLM calls all you need? Towards scaling laws of compound inference systems. *arXiv preprint arXiv:2403.02419*, 2024.
- Narayan, A., Biderman, D., Eyuboglu, E., May, A., Linderman, S., Zou, J., Christopher, R. Minions: Cost-efficient collaboration between on-device and cloud language models. *arXiv preprint arXiv:2502.00030*, 2025.
- Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- Wang, J., Wang, J., Athiwaratkun, B., Zhang, C., and Zou, J. Mixture-of-Agents enhances large language model capabilities. *arXiv preprint arXiv:2406.04692*, 2024.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. NeurIPS*, 2022.
- Zhang, Y., Sun, R., Chen, Y., Pfister, T., Savarese, S., and Xiao, C. Chain of Agents: Large language models collaborating on long-context tasks. *arXiv preprint arXiv:2406.02818*, 2024.

A Supplementary Results

A.1 Benchmark Overview

FinanceBench Domain: SEC financial filings (10-K, 10-Q) Task: Numerical extraction and financial calculation from dense tables Eval: LLM-as-judge — Gemini 2.5 Flash compares predicted vs. gold, outputs CORRECT/INCORRECT. Metric: Accuracy .	<i>150 questions</i> Avg doc: ~10K tokens
LongHealth Domain: Synthetic clinical discharge records Task: Multiple-choice (A–E) medical comprehension, cross-section reasoning Eval: Exact match — extract predicted letter (A–E) vs. gold label. Metric: Accuracy .	<i>400 questions (20 patients × 20 Qs)</i> Avg doc: ~30K tokens
QASPER Domain: NLP research papers (Semantic Scholar) Task: Information-seeking free-form QA over full academic papers Eval: Token F1 — SQuAD-style token overlap, normalized matching. Metric: F1 score .	<i>1,005 questions</i> Avg doc: ~15K tokens

Figure 2: Evaluation benchmarks spanning finance, medicine, and academic domains.

A.2 Performance Recovery

Table 6: Performance recovery (system accuracy / frontier accuracy).

Benchmark	Frontier	MoSA Hybrid	Recovery
FinanceBench	88.0%	60.0%	68.2%
LongHealth	91.0%	74.0%	81.3%
QASPER	51.9%	51.2%	98.6%
Macro Avg	77.0%	61.7%	82.7%

A.3 Architecture Comparison with MinionS

Table 7: Architecture comparison. Both systems use the same 3.2B local model.

Aspect	MinionS	MoSA Hybrid	MoSA SLM
Local model	3B (generic)	3B + 5 QLoRA	3B + 5 QLoRA
Remote model	GPT-4o	DeepSeek V3	None
Remote cost/token	~\$10/M	~\$1.1/M	\$0
Doc routing	Chunks (2–4K)	Full (10–30K)	Full (10–30K)
Fine-tuning	None	QLoRA (6,851)	QLoRA (6,851)

A.4 Cost and Token Efficiency

Table 8: Per-query cost and token statistics for MoSA DeepSeek (benchmark-specific prompts).

Metric	FB ($n=50$)	LH ($n=100$)	QA ($n=100$)
Avg input tokens/query	18,015	108,427	56,550
Avg output tokens/query	2,408	2,888	2,637
Cost/query	\$0.059	\$0.087	\$0.068
Avg time/query	70.0s	102.3s	80.1s
Abstention rate	1.1%	1.8%	1.4%

A.5 Frontier Model Comparison

Table 9: Frontier model comparison. DeepSeek V3 significantly outperforms Kimi K2.5, motivating our adoption of DeepSeek V3 as frontier baseline.

Model	FB	LH	QA (F1)
Kimi K2.5	79.3%	84.8%	29.7%
DeepSeek V3	88.0%	91.0%	51.9%
Δ	+8.7	+6.2	+22.2

A.6 Prompting Strategy Ablation

Table 10: Impact of benchmark-specific vs. generic prompts (MoSA DeepSeek, pre-QLoRA).

Benchmark	Generic	Specific	Δ
FinanceBench	58.0%	58.0%	± 0.0
LongHealth	67.0%	61.0%	-6.0
QASPER	29.5%	38.8%	+9.3
Macro Avg	51.5%	52.6%	+1.1

Benchmark-specific prompting yields +9.3 F1 on QASPER (where academic paper QA requires tailored decomposition) but -6.0pp on LongHealth (where over-constrained medical prompts reduce flexibility on diverse multiple-choice questions).

A.7 QLoRA Training Details

Extractor adapters achieve the lowest loss across all benchmarks (0.164–0.314), consistent with extraction being the most structured task with the most training data. Calculator adapters show the highest loss (0.580–1.166), reflecting both data scarcity and the difficulty of learning arithmetic from language model traces. The LongHealth calculator (9 examples, loss 1.166) is severely undertrained.

Table 11: QLoRA training statistics per agent and benchmark (v1 default: $r=16$, $\alpha=32$, dropout 0.05, 3 epochs, LR 2×10^{-4}). Sorted by final loss.

Agent	Benchmark	Train Examples	Final Loss
Extractor	QASPER	2,881	0.164
Extractor	FinanceBench	294	0.255
Extractor	LongHealth	549	0.314
Orchestrator	FinanceBench	98	0.343
Orchestrator	QASPER	1,065	0.492
Orchestrator	LongHealth	247	0.531
Calculator	FinanceBench	82	0.580
Synthesizer	QASPER	987	0.610
Synthesizer	LongHealth	241	0.741
Reasoner	LongHealth	228	0.752
Reasoner	FinanceBench	45	0.752
Calculator	QASPER	64	0.887
Synthesizer	FinanceBench	100	0.899
Reasoner	QASPER	394	0.896
Calculator	LongHealth	9	1.166

Table 12: FinanceBench accuracy by question type, pre- and post-QLoRA. SLM gains concentrate in calculation (+13pp) and numeric extraction (+9pp); Hybrid gains in yes/no (+17pp) and reasoning (+40pp).

Type	n	Hyb. Pre	Hyb. QLoRA	SLM Pre	SLM QLoRA
Calculation	15	46.7	46.7	6.7	20.0
Extract (numeric)	11	54.5	54.5	18.2	27.3
Extract (qualitative)	9	66.7	55.6	22.2	33.3
Yes/No	6	66.7	83.3	50.0	33.3
Reasoning	5	60.0	100	80.0	60.0

A.8 FinanceBench Question-Type Breakdown

SLM QLoRA gains are concentrated in calculation and extraction tasks, where the fine-tuned orchestrator learned multi-step decomposition. Hybrid QLoRA gains are on yes/no and reasoning tasks, where reduced reasoner abstention (45% $\rightarrow$ 5%) pays off when paired with a strong synthesizer.

A.9 Confidence Calibration

Table 13: Average confidence on correct vs. incorrect answers. Larger gaps indicate better calibration.

Benchmark	Config	Correct	Wrong	Gap
FinanceBench	Pre-QLoRA Hybrid	0.95	0.79	0.157
FinanceBench	QLoRA Hybrid	0.95	0.88	0.070
LongHealth	QLoRA Hybrid	0.96	0.88	0.070
LongHealth	QLoRA SLM	0.90	0.79	0.109

QLoRA fine-tuning degrades calibration on FinanceBench (gap shrinks from 0.157 to 0.070—the model becomes overconfident on wrong answers) but maintains reasonable calibration on LongHealth. The SLM

shows the best calibration (0.109 gap)—it reliably signals uncertainty—but the 3B synthesizer cannot leverage this signal for error recovery.

A.10 Latency and Cost Comparison (QLoRA vs. Pre-QLoRA)

Table 14: Per-query latency and cost, pre- vs. post-QLoRA. QLoRA increases latency $1.4\text{--}2.5\times$ due to longer model outputs.

Benchmark	Config	Pre (s)	QLoRA (s)	Pre (\$)	QLoRA (\$)
FinanceBench	Hybrid	70.0	166.6	0.059	0.141
FinanceBench	SLM	66.4	111.5	0.056	0.094
LongHealth	Hybrid	102.3	144.4	0.087	0.122
LongHealth	SLM	85.9	216.0	0.073	0.183
QASPER	Hybrid	80.1	109.3	0.068	0.093
QASPER	SLM	32.3	74.0	0.027	0.063

QLoRA models generate longer outputs, increasing both latency and cost. The largest slowdown is LongHealth SLM ($2.5\times$), driven by 46% more agent calls from the QLoRA orchestrator’s aggressive decomposition and 98% more output tokens per query. Despite higher per-query costs, QLoRA Hybrid’s best return on investment is QASPER: +12.4 F1 for only 37% more cost.

B Agent Prompts

Below we provide representative prompts for key agents. All prompts were written by the authors.

B.1 Orchestrator Prompt (FinanceBench)

You are a task decomposition agent. Given a financial question about a company filing, break it into sub-tasks for specialist agents.

Available specialists:

extractor -- lookup a single number, fact, or line item
calculator -- arithmetic: ratios, percentages, year-over-year changes
reasoner -- judgment or comparison across multiple facts

Rules:

1. Use as many sub-questions as needed. A ratio calculation needs extractors for each operand PLUS a calculator step.
2. If the question contains a formula, create extractors for each operand and a calculator for the formula.
3. Calculator text must include the formula and reference which sq values to use.

Few-shot example:

Q: "What is the FY2017 ROA for Coca Cola?"
-> sq1 (extractor): net income FY2017
-> sq2 (extractor): total assets FY2016 and FY2017
-> sq3 (calculator): $ROA = sq1 / ((sq2_{2016} + sq2_{2017}) / 2)$

B.2 Extractor Prompt (Financial Domain)

You are a careful financial document reader. Extract the requested information from the document chunk.

ANSWER FORMAT RULES:

- Percentages: return as a number (e.g. "23.5" not "23.5%")
- Numbers: return bare number only (e.g. "8.74" not "\$8,738M")
- Names: list ALL specific names, separated by commas
- IMPORTANT: keep extracted_text SHORT -- max 2 lines

FINANCIAL EXTRACTION GUIDANCE:

- Match the EXACT line item asked about
- CHECK UNITS: if "in USD billions" and doc shows millions, divide by 1000
- Navigate: Income Statement, Balance Sheet, Cash Flow Statement

B.3 Synthesizer Prompt (Shared Across Benchmarks)

Critically evaluate the agent results and synthesize a single final answer. Some agents may be wrong.

EVALUATION RULES:

- Compare agent evidence directly, not confidence scores
- Check the Citation field -- trust agents with strong citations
- If agents conflict, trust the one whose citation best matches
- If arithmetic is needed and no agent computed it, do it yourself in scratchpad

Think step-by-step in the "scratchpad" field before giving your answer.

```
{"response": "<final answer>", "confidence": <0.0-1.0>,
 "scratchpad": "<step-by-step reasoning>"}
```

C Detailed Results

Table 15: Detailed per-benchmark metrics for MoSA DeepSeek (benchmark-specific prompts).

Metric	FinanceBench	LongHealth	QASPER
Score	58.0%	61.0%	38.8% (F1)
Eval samples	50	100	100
Avg time/query	70.0s	102.3s	80.1s
Avg input tokens	18,015	108,427	56,550
Avg output tokens	2,408	2,888	2,637
Cost/query	\$0.059	\$0.087	\$0.068
Total cost	\$2.97	\$8.67	\$6.78
Samples generated	157	270	309
Abstention rate	1.1%	1.8%	1.4%

D Frontier Model Comparison

Following TA feedback on our choice of Kimi K2.5 as frontier model at milestone, we compared two frontier models:

Table 16: Frontier model comparison: DeepSeek V3 vs. Kimi K2.5.

Model	FinanceBench	LongHealth	QASPER (F1)
Kimi K2.5 (128K)	79.3%	84.8%	29.7%
DeepSeek V3	88.0%	91.0%	51.9%
Δ	+8.7	+6.2	+22.2

DeepSeek V3 significantly outperforms Kimi K2.5 on all three benchmarks, particularly on QASPER where the gap is +22.2 F1 points. We adopted DeepSeek V3 as our primary frontier baseline because it provides a more competitive upper bound and is more cost-effective (\$0.14/M vs. \$0.60/M input tokens).