Experience-Guided Behavior Adaptation for Large Language Models

Iknoor Singh [1] **Harjot Singh** [1] **Abhishek Tripathi** [1] **Niresh Agarwal** [1] **Murat Sensoy** [1]

Abstract

Large language models (LLMs) cannot accumulate experience across interactions without parameter updates. Retrieval-augmented generation and memory-based approaches attempt to leverage past interactions but typically rely on semantic similarity alone and ignore whether experiences actually improve performance. We introduce an uncertainty-aware guidance framework that distills compact guidance from past failures and selects it via a contextual bandit formulation. Each guidance item maintains a Beta posterior over effectiveness, and Thompson sampling balances exploration and exploitation, allowing the model to downweight unhelpful guidance over time. Across benchmarks, our method corrects up to 69.5% of prior errors and improves Haiku 4.5 accuracy by up to 26%. Notably, guidance distilled from a weaker open-weight model (Qwen3 4B) transfers effectively to a stronger proprietary model (Haiku 4.5), demonstrating experience exchange across models in the context space.

1. Introduction

Large language models exhibit remarkable in-context learning abilities (Brown et al., 2020; Dong et al., 2022), adapting to new tasks through examples or instructions without parameter updates (Ouyang et al., 2022). However, LLMs remain fundamentally stateless: they cannot consolidate experience across interactions, causing repeated errors across sessions (Wang et al., 2024; Zhuo et al., 2025). Retrieval-augmented generation (Lewis et al., 2020) and recent memory frameworks (Chhikara et al., 2025; Ouyang et al., 2025) aim to leverage past experience, but existing approaches either rely primarily on semantic similarity (Ouyang et al., 2025) or assume a bounded set of insights that can be directly included in the context (Suzgun et al., 2025). These assumptions break down as insights accumulate, making it increasingly difficult to determine which insights are actually useful. This raises a key question: *How can we guide LLMs at scale using empirically validated insights?*

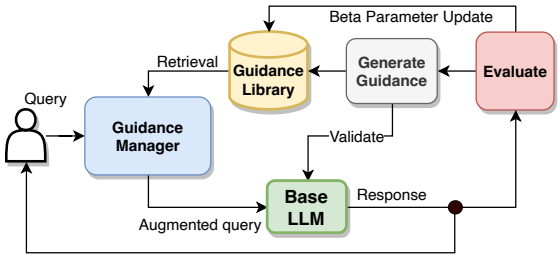

Figure 1. Overview of Experience-Guided Behavior Adaptation.

We propose *Experience-Guided Behavior Adaptation*, where LLMs improve by learning in the context space through structured guidance distilled from failures (Figure 1). Here, each guidance is treated as a *hypothesis* whose utility is validated through experience. Guidance is created retrospectively from past interactions and its effectiveness is tracked over time. We address this by: (1) modeling each guidance item as a contextual bandit arm, (2) maintaining a Bayesian posterior over its success probability using a Beta distribution, and (3) selecting guidance via a Thompson-sampling reranking that balances semantic relevance with posterior reliability. This approach naturally supports exploration, exploitation, and *forgetting* of ineffective guidance. Moreover, this enables continuous behavioral improvement while keeping parameters frozen.

2. Experience-Guided Behavior Adaptation

We introduce a framework for improving language models by augmenting context with reusable guidance, referred to as *contextual guidance*. Here, a contextual guidance is defined as a tuple $g = (guidance, minimal\ example)$, where *guidance* provides generalizable, actionable advice for solving related problems and the optional *minimal example* illustrates its application in practice.

2.1. Problem Setup

We consider a frozen LLM π_θ mapping context C to outputs $Y \sim \pi_\theta(\cdot \mid C)$. Context augmentations ΔC can improve

[1]Amazon, London, UK. Correspondence to: Iknoor Singh <iknoor@amazon.co.uk>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

outputs on a given request, but instance-specific augmentations do not transfer to future requests. We study *reusable contextual guidance*: augmentations distilled from past interactions that generalize across future queries while accounting for uncertainty in their effectiveness.

2.2. Guidance Manager

We introduce the guidance manager, a policy that maintains a library of contextual guidance items. It continuously updates the guidance library with new experiences and feedback. The guidance library is formalized as follows:

$$\mathcal{L} = \{(g_i, \alpha_i, \beta_i)\}_{i=1}^{N},$$

where (α_i, β_i) are Beta distribution parameters tracking empirical success and failure counts. Each item g_i acts as a stochastic policy component with mean success probability

$$\hat{p}_i = \mathbb{E}[\text{Beta}(\alpha_i, \beta_i)] = \frac{\alpha_i}{\alpha_i + \beta_i}.$$

Guidance Generation. Given a sub-optimal output Y for context C with feedback F, a frozen supervisor π_ϕ generates candidate guidance: $g \sim \pi_\phi(g \mid C, F)$. Each candidate is validated by evaluating $\pi_\theta(\cdot \mid C \oplus g)$ before being added to the library with a uniform Beta prior ($\alpha = 1, \beta = 1$). See Appendix A.1–A.2 for prompt and examples.

Updating Guidance Policy. Each guidance item g has a latent success probability $p_g \in [0, 1]$, capturing its effectiveness in relevant contexts. We use Beta prior on p_g:

$$p_g \sim \text{Beta}(\alpha_g^{(0)}, \beta_g^{(0)}), \quad \alpha_g^{(0)} = \beta_g^{(0)} = 1. \tag{1}$$

Whenever g is applied, we observe a binary outcome $y_g \in \{0, 1\}$ from the environment (ground truth or judge), where $y_g = 1$ indicates success and $y_g = 0$ failure. Let s and f denote successes and failures; the likelihood is $p_g^s(1 - p_g)^f$, and the prior is updated via Bayes' theorem:

$$\alpha_g = \alpha_g^{(0)} + s, \quad \beta_g = \beta_g^{(0)} + f. \tag{2}$$

This conjugate update maintains a posterior over each guidance item's effectiveness, enabling principled exploration–exploitation via Thompson sampling.

Guidance Retrieval. We frame guidance selection as a contextual bandit with a dynamically growing action space. We adopt a two-stage retrieve-then-rerank scheme that decouples coarse recall from fine-grained evaluation (Han et al., 2024; Zhou & Mannor, 2025). Each query q and item g are encoded as embeddings $\mathbf{e}_q, \mathbf{e}_g$, and a top-$K$ candidate set $\mathcal{L}_K(q)$ is retrieved by cosine similarity. We re-rank via Thompson sampling, drawing $\tilde{p}_g \sim \text{Beta}(\alpha_g, \beta_g)$ for each $g \in \mathcal{L}_K(q)$ and score:

$$\text{score}(q, g) = \text{cosine}(\mathbf{e}_q, \mathbf{e}_g) \cdot \tilde{p}_g. \tag{3}$$

The cosine term enforces semantic relevance while the posterior down-weights ineffective guidance, yielding implicit forgetting without explicit pruning. We use a fixed guidance library in all experiments; a Bayesian pruning strategy for lifelong settings is discussed in Appendix A.5.

3. Experimental Evaluation

Datasets. We evaluate on: **(1) MMLU-Pro** (Wang et al., 2024) — six domain subsets (Physics, Engineering, Mathematics, Chemistry, Law, Economics), 500 questions each; **(2) GPQA** (Rein et al., 2024) — GPQA-Main (448 graduate-level questions) and GPQA-Diamond (198 expert-validated); **(3) AIME 2020–2024** (Veeraboina, 2023) — 133 maths problems used to assess guidance transferability; and **(4) MBPP** (Austin et al., 2021) — 374 training and 90 validation Python programming tasks. Overall, our evaluation spans 11 datasets across multiple knowledge domains.

Models. We evaluate open-weight Qwen3 4B (Yang et al., 2025) and the proprietary Haiku 4.5 (Anthropic, 2025). Claude Opus 4.5 serves as the supervisor model for guidance generation. We use Cohere V3 embeddings (Cohere, 2025) and ANN for vector search (Arya et al., 1998).

Evaluation. We evaluate under two modes: **(i) Progressive Guidance Generation Mode:** Evaluates whether guidance from past failures can correct future errors. We iteratively build a guidance library in temporal order from failures of frozen π_θ. For each input C, the model produces $Y \sim \pi_\theta(\cdot \mid C)$. If Y is incorrect, relevant guidance is retrieved via Thompson sampling (Section 2.2). A failed query can only be recovered using guidance from *previous* failures. This enforces strict temporal causality. If the retrieved guidance fails to produce a correct output, the supervisor generates new candidate guidance items. New guidance is *validated* before being added to the library with a uniform Beta prior. Beta parameters are updated based on observed outcomes, accumulating empirical evidence of effectiveness. **(ii) Frozen Guidance Mode:** Evaluates generalization under deployment constraints. The guidance library is fixed at inference time. Guidance is retrieved via Thompson sampling and applied to every query; no new guidance is generated or updated. In this, we reuse the guidance library from progressive generation mode and evaluate it on disjoint samples. For eg., in MMLU-Pro, the first 500 samples per domain are used for guidance construction and the remaining samples for evaluation. For smaller datasets, we transfer guidance across related benchmarks (e.g., AIME uses guidance from MMLU-Pro Math; GPQA

Table 1. Progressive guidance generation mode. *Recovery Rate* is the fraction of initial failures corrected after applying retrieved guidance.

Dataset	Qwen3 4B				Haiku 4.5			
	Baseline	RAG	Ours	Recovery Rate	Baseline	RAG	Ours	Recovery Rate
MMLU Pro Physics	36.8%	27.2%	**47.0%**	16.1%	64.6%	69.6%	**89.2%**	69.5%
MMLU Pro Engineering	40.2%	26.2%	**49.0%**	14.7%	59.2%	58.0%	**78.0%**	46.1%
MMLU Pro Maths	38.4%	24.0%	**48.8%**	16.9%	71.2%	72.2%	**90.0%**	65.3%
MMLU Pro Chemistry	33.6%	25.4%	**45.4%**	17.8%	58.2%	63.2%	**86.0%**	66.5%
MMLU Pro Law	27.2%	4.0%	**35.2%**	11.0%	57.4%	59.2%	**70.6%**	31.0%
MMLU Pro Economics	60.2%	29.4%	**65.4%**	13.1%	78.0%	76.6%	**87.8%**	44.5%
MMLU Pro Bio	70.6%	46.2%	**74.4%**	12.9%	85.2%	85.6%	**91.6%**	43.2%
GPQA Main (w/o Diamond)	30.0%	16.4%	**33.6%**	5.1%	47.6%	48.4%	**72.8%**	48.1%
MBPP – Train Set	67.1%	64.7%	**73.0%**	17.9%	88.8%	86.9%	**90.1%**	11.9%

Table 2. Frozen guidance mode. Baselines are computed on disjoint splits and thus differ from those in Table 1. Improvement is the absolute gain in performance.

Dataset	Qwen3 4B			Haiku 4.5			Guidance Generation
	Baseline	Ours	Improvement	Baseline	Ours	Improvement	
MMLU Pro Physics (500)	29.8%	**33.4%**	+3.6%	60.2%	**82.6%**	+22.4%	First 500 samples
MMLU Pro Engineering (469)	44.8%	**45.0%**	+0.2%	56.7%	**65.5%**	+8.8%	First 500 samples
MMLU Pro Maths (500)	31.4%	**38.0%**	+6.6%	73.3%	**91.9%**	+18.6%	First 500 samples
MMLU Pro Chemistry (500)	32.4%	**37.8%**	+5.4%	58.6%	**84.7%**	+26.1%	First 500 samples
MMLU Pro Law (500)	28.2%	**30.6%**	+2.4%	54.8%	**58.4%**	+3.6%	First 500 samples
MMLU Pro Economics (344)	60.2%	**62.5%**	+2.3%	79.0%	**88.0%**	+9.0%	First 500 samples
MMLU Pro Bio (217)	76.5%	**78.3%**	+1.8%	91.2%	**91.2%**	+0.0%	First 500 samples
AIME 2020–2024 (133)	4.9%	**7.3%**	+2.4%	51.1%	**51.9%**	+0.8%	MMLU-Pro Math
GPQA Diamond (198)	**35.4%**	**35.4%**	+0.0%	46.5%	**63.1%**	+16.6%	GPQA (w/o Diamond)
MBPP – Validation Set (90)	56.7%	63.3%	+6.7%	**86.7%**	**86.7%**	+0.0%	MBPP train set

Diamond uses guidance from disjoint GPQA Main samples). This setup tests if guidance captures reusable, general reasoning patterns rather than instance-specific solutions.

Evaluation Metrics. **(i) Success Rate:** proportion of tasks completed successfully. **(ii) Recovery Rate:** fraction of initially failed tasks corrected when guidance is applied: $|\mathcal{R}|/|\mathcal{F}|$, where $\mathcal{F}$ is the set of failures and $\mathcal{R} \subseteq \mathcal{F}$ those that succeed with guidance.

4. Results and Discussion

4.1. Progressive Guidance Generation

Table 1 reports results against direct LLM generation (*baseline*) and 5-shot *RAG* over similar training queries and their ground-truth answers. We fix $K = 5$ guidance items per context (Appendix A.4). We find that Qwen3 4B consistently exhibits lower recovery rates (5.1%–17.9%) versus Haiku 4.5 (11.9%–69.5%), reflecting the weaker model's limited capacity to utilize guidance. The RAG baseline degrades Qwen3 4B performance substantially on several tasks (e.g., MMLU-Pro Economics, Biology), while Haiku 4.5 shows mixed but less severe behavior. Overall, progressive guidance generation enables test-time improvement by distilling failure-specific information into reusable guidance, outperforming RAG on all benchmarks for both models. Please refer to Appendix A.3 for the number of guidance items generated across datasets.

4.2. Frozen Guidance Library

Table 2 shows results with a frozen guidance library. Haiku 4.5 achieves large absolute gains: 26.1% on MMLU-Pro Chemistry, 22.4% on Physics, and 16.6% on GPQA Diamond. Qwen3 4B shows more modest but consistently positive gains (up to 6.7% on MBPP). Cross-benchmark generalization is also observed: guidance from MMLU-Pro Maths improves AIME (+2.4% for Qwen3 4B, +0.8% for Haiku 4.5). When baseline accuracy is already high (>86%), improvement headroom is limited, as expected. These results confirm that once constructed, the guidance library enables test-time gains with minimal overhead, fully decoupled from guidance generation.

4.3. Cross-Model Guidance Transfer

We investigate cross-model guidance transfer by guiding the stronger closed-weight Haiku 4.5 with guidance constructed from the weaker open-weight Qwen3 4B (Table 3). Gains are substantial on several MMLU-Pro subsets: $+25.1\%$ on Chemistry, $+23.2\%$ on Physics, $+18.6\%$ on Maths, and $+15.9\%$ on GPQA Diamond – comparable to guidance learned directly from Haiku 4.5 failures (Table 2). Per-

Table 3. Performance of Haiku 4.5 with guidance learned from Qwen3 4B in frozen guidance mode.

Dataset	Haiku 4.5			Guidance Generation (Qwen3 4B)
	Baseline	Ours	Improvement	
MMLU Pro Physics (500)	60.2%	**83.4%**	+23.2%	First 500 samples
MMLU Pro Engineering (467)	56.7%	**69.2%**	+12.5%	First 500 samples
MMLU Pro Maths (500)	73.3%	**91.9%**	+18.6%	First 500 samples
MMLU Pro Chemistry (500)	58.6%	**83.7%**	+25.1%	First 500 samples
MMLU Pro Law (500)	54.8%	**58.2%**	+3.4%	First 500 samples
MMLU Pro Economics (343)	79.0%	**88.0%**	+9.0%	First 500 samples
MMLU Pro Bio (217)	**91.2%**	90.8%	−0.4%	First 500 samples
AIME 2020–2024 (133)	51.1%	**52.3%**	+1.2%	MMLU-Pro Math
GPQA Diamond (198)	46.5%	**62.4%**	+15.9%	GPQA (w/o Diamond)
MBPP – Validation Set (90)	**86.7%**	**86.7%**	+0.0%	MBPP train set

Table 4. Effect of Thompson sampling on Haiku 4.5.

Dataset	Baseline	Retrieval	TS_{pg}	TS_{10}
MMLU Pro Phy.	60.2%	83.2%	82.6%	**84.3%**
MMLU Pro Eng.	56.7%	65.7%	65.5%	**67.0%**
MMLU Pro Maths	73.3%	91.0%	**91.9%**	91.6%
MMLU Pro Chem.	58.6%	84.1%	84.7%	**84.9%**
MMLU Pro Law	54.8%	57.0%	58.4%	**60.8%**
MMLU Pro Eco.	79.0%	87.5%	88.0%	**89.2%**
MMLU Pro Bio	91.2%	89.9%	91.2%	**92.2%**
GPQA Diamond	46.5%	61.1%	63.1%	**65.2%**

formance gains are smaller where Haiku 4.5 already excels. These results show that guidance from weaker models generalizes to stronger ones, capturing model-agnostic reasoning patterns and enabling shared experience repositories.

4.4. Impact of Thompson Sampling for Re-ranking

Thompson Sampling (TS) provides a principled exploration–exploitation mechanism under posterior uncertainty. However, in the progressive generation mode (Section 3), guidance items are created sequentially from failures and only become available for future queries. As a result, exposure is highly imbalanced: early guidance is evaluated frequently, while later items receive limited trials. This induces sparse and uneven evidence of the quality of guidance items, with many posteriors remaining close to their priors.

To better evaluate TS under reliable posterior estimates, we introduce TS_{10}: each guidance item receives ten evaluation trials on semantically relevant queries, updating success/failure counts. We then evaluate on disjoint set with the frozen library to measure whether TS improves selection. Baseline and TS_{pg} (TS re-ranking using posteriors from progressive generation) are copied from Table 2. We also report guidance retrieval without TS re-ranking (*Retrieval*). While retrieval yields strong gains and TS_{pg} shows only marginal improvements due to sparse evidence (Table 4). In contrast, TS_{10} consistently improves performance and achieves the best results. This indicates that TS becomes effective when posteriors better reflect guidance reliability.

5. Related Work

LLMs adapt to new tasks via in-context learning without parameter updates (Brown et al., 2020). Prompt optimization refines prompts (Zhao et al., 2023; Pryzant et al., 2023), and iterative methods use self-consistency (Wang et al., 2022), reflection (Shinn et al., 2023), and self-refinement (Madaan et al., 2023). However, these methods are session-bound and do not retain experience across interactions.

ReasoningBank (Ouyang et al., 2025), Expel (Zhao et al., 2024), and Dynamic Cheatsheet (DC) (Suzgun et al., 2025) accumulate experience to improve test-time performance without parameter updates, but rely on embedding similarity for retrieval—assuming semantic relevance implies utility. We conducted additional experiments comparing DC with our approach on same subset of MMLU-Pro Phys & Eng. We find that DC achieves 83.8% and 60.0%, respectively, whereas our method achieves 89.2% and 78.0%. Moreover, DC requires costly per-query synthesis which increases inference overhead. Overall, our approach differs in three ways: (1) we validate guidance before storage to prevent accumulation of ineffective guidance items (a limitation noted in ExpeL), (2) we maintain Beta posteriors over empirical success to down-weight unhelpful but semantically relevant guidance, and (3) we support frozen-library deployment and cross-model transfer, not studied in prior work.

6. Conclusions

We presented *Experience-Guided Behavior Adaptation*, enabling frozen LLMs to improve from past failures without parameter updates. By formulating guidance selection as a contextual bandit with Thompson sampling over Beta posteriors, the approach achieves substantial gains over direct generation and RAG baselines. The frozen guidance library yields substantial gains, motivating a clean separation between offline guidance construction and online inference. Cross-model guidance transfer shows that distilled guidance captures transferable task-level reasoning patterns, enabling shared experience repositories across model families.

7. Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. We study a framework that enables frozen language models to reuse experience from past failures at test time without parameter updates, with guidance generated and reviewed offline before reuse. There are many potential societal consequences of this line of research, none of which we feel must be specifically highlighted here.

References

Anthropic. Claude: A family of large language models. https://www.anthropic.com/, 2025. Accessed: 2026-01-12.

Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

Austin, J., Odena, A., Nye, M., Bosma, M., Michalewski, H., Dohan, D., Jiang, E., Cai, C., Terry, M., Le, Q., and Sutton, C. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chhikara, P., Khant, D., Aryan, S., Singh, T., and Yadav, D. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Cohere. Cohere embed models. https://docs.cohere.com/docs/cohere-embed, 2025. Accessed: 2026-01-12.

Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., Xia, H., Xu, J., Wu, Z., Liu, T., et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Han, S., Lakritz, R., and Wu, H. Augmented two-stage bandit framework: Practical approaches for improved online ad selection. In *ADKDD*, 2024.

Kaufmann, E., Cappe, O., and Garivier, A. On bayesian upper confidence bounds for bandit problems. In Lawrence, N. D. and Girolami, M. (eds.), *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pp. 592–600, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL https://proceedings.mlr.press/v22/kaufmann12.html.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Ouyang, S., Yan, J., Hsu, I., Chen, Y., Jiang, K., Wang, Z., Han, R., Le, L. T., Daruki, S., Tang, X., et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025.

Pryzant, R., Iter, D., Li, J., Lee, Y. T., Zhu, C., and Zeng, M. Automatic prompt optimization with" gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Suzgun, M., Yuksekgonul, M., Bianchi, F., Jurafsky, D., and Zou, J. Dynamic cheatsheet: Test-time learning with adaptive memory. *arXiv preprint arXiv:2504.07952*, 2025.

Veeraboina, H. Aime problem set 1983-2024, 2023. URL https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024.

Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., Chowdhery, A., and Zhou, D. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., et al. Mmlu-pro:

A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhao, A., Huang, D., Xu, Q., Lin, M., Liu, Y.-J., and Huang, G. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Zhou, Q. and Mannor, S. Representative action selection for large action space: From bandits to mdps. 2025.

Zhuo, T. Y., He, J., Sun, J., Xing, Z., Lo, D., Grundy, J., and Du, X. Identifying and mitigating api misuse in large language models. *arXiv preprint arXiv:2503.22821*, 2025.

A. Appendix

A.1. System Prompts

This section presents the system prompts used throughout our framework. We use three categories of prompts:

- **Guidance Generation** (Figure 2): Used by the supervisor model to extract reusable guidance from failed attempts. The prompt instructs the model to perform root cause analysis and generate actionable, self-contained guidance with example applications.
- **Baseline Generation** (Figures 3, 4): Standard prompts for code generation and question answering tasks without any guidance context. These establish baseline model performance.
- **Validation with Guidance Context** (Figures 5, 6): Extended prompts that inject retrieved guidance into context and require the model to report which guidance items were actually applied via `used_guidance_ids`. This enables tracking of guidance utilization for updating Beta posteriors in our bandit framework.

All prompts are held constant across experiments to ensure consistent evaluation. The `{additional_guidance_context}` placeholder is populated at runtime with retrieved guidance items formatted with unique IDs.

Guidance Generation Prompt
You are a **Supervisory LLM** specialized in generating guidance. Your role is to extract reusable guidance from failed task attempts and generate **actionable guidance** to help prevent similar reasoning errors in future problem-solving.
Analysis Process:

1. **Root Cause Analysis**: Identify the specific cause that led to the incorrect answer.
2. **Generalizable Guidance**: Derive concrete, generalizable guidance that applies broadly to similar problems.
3. **Self-Contained Guidance**: Ensure each guidance is unique, independent, and immediately actionable without relying on other guidance.

Output Requirements: Generate **1–3 independent guidance objects**. Each guidance object MUST include:

1. **guidance**: Clear, actionable guidance for solving similar problems.
2. **example application**: A brief example demonstrating how to apply this guidance.

Figure 2. Guidance Generation System Prompt

Python Code Generation Prompt

You are a Python coding assistant. Write complete, correct, working, and efficient Python function code to solve the given problem.

Output Format:

```
{"code": "your_python_function_here"}
```

Constraints:

- Do **not** include explanations.
- Do **not** include markdown formatting.
- Do **not** include any additional text.

Figure 3. System prompt used for Python code generation.

Question Answer Generation Prompt

You are a highly knowledgeable expert across all academic domains. You will be given a multiple-choice question.

Analyze the question carefully and reason step by step using relevant facts and logical deductions. Then, select the single best answer from the provided options.

Required Output:

- **answer:** The single letter (A, B, C, D, etc.) corresponding to the chosen option.
- **explanation:** A brief step-by-step explanation justifying the selected answer.

Figure 4. System prompt used for question answering.

Code Generation with Guidance Context

You are a Python coding assistant. Write complete, correct, working and efficient Python function code to solve the given problem. IMPORTANT: The following are guidance from past experience. If any applies, follow its guidance and mention its ID in final response.

`{retrieved_guidance_context}`

IMPORTANT: If you use any of the guidance provided above, you MUST include their guidance IDs in the `used_guidance_ids` field. Only include IDs of guidance that you actually applied. If no guidance was used, set `used_guidance_ids` to an empty array `[]`.

Output format:

```
{"code": "your_python_function_here", "used_guidance_ids": ["id1", "id2"]}
```

Figure 5. Validation prompt for code generation with retrieved guidance context.

Question Answering with Guidance Context

You are a highly knowledgeable expert across all academic domains. You will be given a multiple-choice question. Analyze the question and think step by step about relevant facts, reasoning, and possible options. Then, choose the single best answer from the provided options.
IMPORTANT: The following are guidance items from past experience. If any applies, follow its guidance and mention its ID in final response.

`{retrieved_guidance_context}`

You MUST provide:

1. **answer**: The single letter (A, B, C, D, etc.) corresponding to your chosen answer
2. **explanation**: A brief step-by-step explanation of why you chose this answer
3. **used_guidance_ids**: IDs of guidance you actually applied (empty array `[]` if none used)

Figure 6. Validation prompt for question answering with retrieved guidance context.

Example 1: MBPP (Code Generation)

Query: Write a function to find average value of the numbers in a given tuple of tuples. Test cases: ....

Initial Error: Model computed a single global average instead of column-wise averages.

Applied Guidance: When solving a problem, first infer the *structural alignment* between the input and the expected output. Determine which dimension, grouping, or axis of the input corresponds to each element of the output, and ensure operations (e.g., aggregation, transformation, comparison) are applied along that structure rather than over the entire input indiscriminately.

Minimal Example:

```
Input:  [[1, 2, 3], [4, 5, 6]]
Expected:  [6, 15] (sum of each row)
Wrong approach:  Sum all elements = 21
Correct approach:  [sum([1,2,3]), sum([4,5,6])]
```

Figure 7. Guidance corrects misinterpretation of output structure in code generation.

Example 2: MMLU-Pro Law

Query: A detective found a footprint from a left-foot shoe at a murder scene. The print was preserved appropriately as evidence. It had distinctive tread marks and an unusual wear pattern on the sole. It also had a "V" mark on the heel bottom that indicates the brand was a Victory shoe. The detective, armed with a proper search warrant, searched the suspect's apartment, where he found a shoe to a right foot that of the same size, and with a similar wear pattern and the same "V" mark as the shoeprint found at the scene. The shoe for the left foot was not found but the shoe for the right foot was seized from the suspect's closet and offered as evidence at trial. Is this admissible evidence?

Initial Error: Model selected a factually relevant but legally imprecise answer.

Applied Guidance: Avoid selecting answers that are merely 'close' or factually accurate when a more legally precise answer exists. An answer describing a circumstance (like retreat) may be true but secondary to an answer stating the controlling legal rule (like proportionality).

Minimal Example:

> When evaluating 'No, because he was leaving' vs. 'No, because deadly force was unreasonable for theft,' recognize that the first describes a fact while the second states the legal principle. The principle-based answer is typically correct.

Figure 8. Guidance helps distinguish between factually accurate and legally precise answers.

Example 3: MMLU-Pro Engineering

Query: What will be the number of lamps, each having 300 lumens, required to obtain an average illuminance of 50 lux on a 4m × 3m rectangular room?

Initial Error: The model introduced additional correction or efficiency factors that were not specified in the problem statement, resulting in an incorrect or unnecessarily complex solution.

Applied Guidance: When an engineering or applied physics problem provides only fundamental quantities and does not explicitly mention correction factors (e.g., utilization factors, efficiency losses, maintenance factors, safety margins), the solution should be based solely on the direct theoretical relationship between the given variables. Additional factors should be introduced only if they are explicitly stated or if the problem clearly requests a practical or real-world estimate.

Minimal Example:

> Suppose a problem asks for the number of identical units N, each contributing an amount u, to achieve a required total effect T over a system of size S, where the defining relationship is
>
> $$T = \frac{N \cdot u}{S}.$$
>
> Use only the quantities explicitly provided in the problem, and do not introduce additional correction factors unless they are stated.

Figure 9. General guidance demonstrating correct application of direct formulas without unstated correction factors.

A.2. Examples

Figures 7–8 are the representative examples illustrating how retrieved guidance corrects model errors across different tasks. These examples illustrate that guidance captures reusable reasoning patterns—such as interpreting output structure, selecting legally precise answers, and avoiding unnecessary assumptions that generalize across related problems within each domain.

A.3. Guidance Items Comparison

Figure 10 shows the number of guidance items generated across datasets in progressive guidance generation mode. Qwen3 4B consistently produces more guidance than Haiku 4.5, reflecting its higher failure rate.

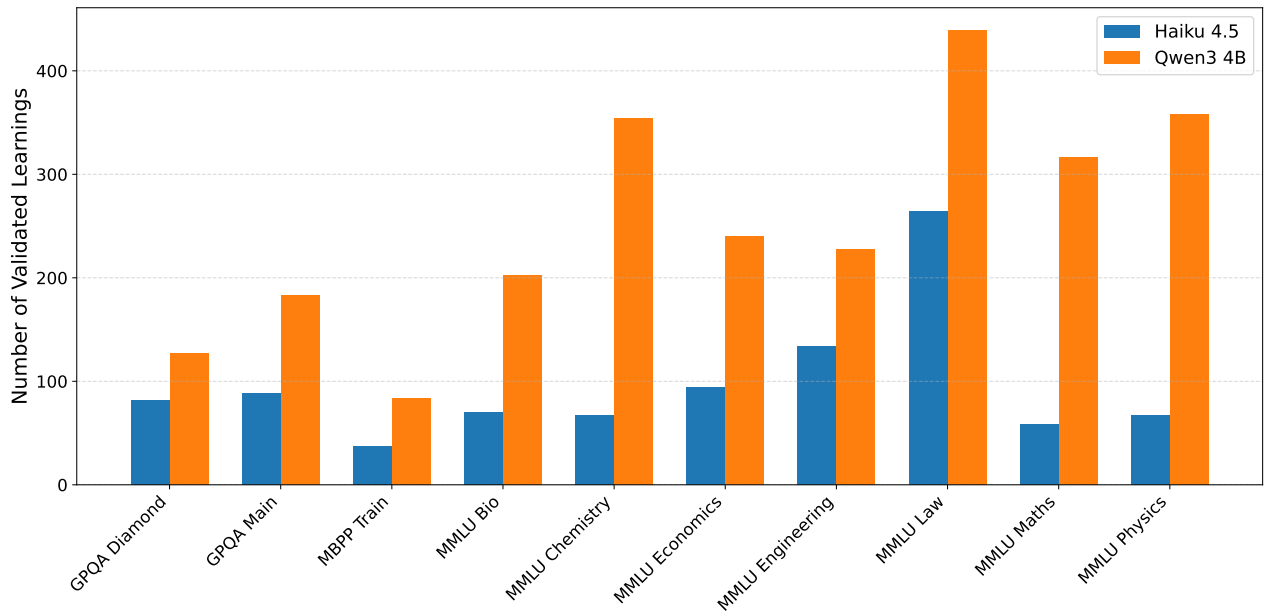

Figure 10. Number of guidance items across different datasets.

A.4. Effect of Number of Guidance Items

We analyze the effect of the number of guidance items included in the context. Our pipeline retrieves top ten relevant candidates via vector search, re-ranks them using Thompson sampling, and selects the top $\{1, 3, 5, 7\}$ items for context augmentation. Figure 11 shows results on MMLU-Pro Engineering and Law using Haiku 4.5.

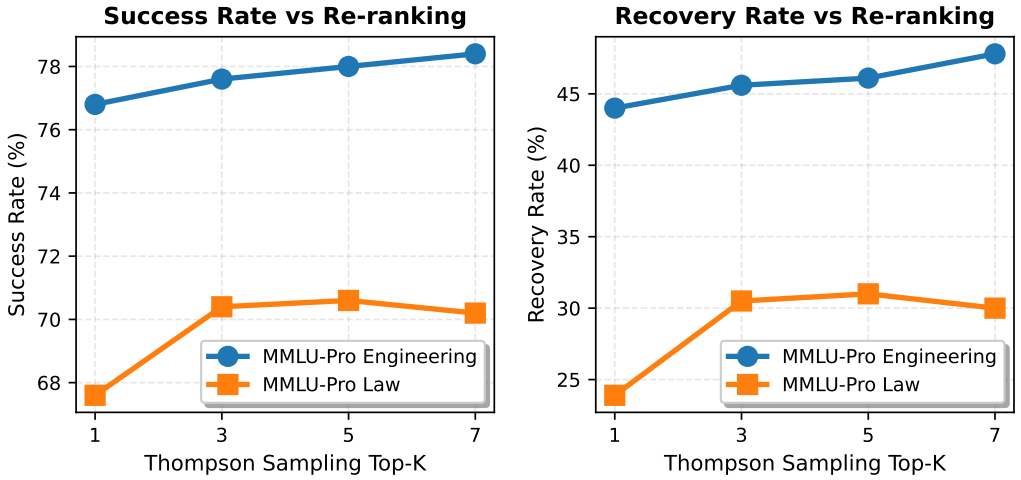

Figure 11. Effect of number of guidance items (Haiku 4.5).

Increasing the number of guidance items improves performance, with gains saturating around five items. The recovery rate increases more sharply than the overall success rate, indicating that correcting a failure often requires multiple guidance items. Based on this trend, we fix $K = 5$ for all main experiments, for both progressive guidance generation and frozen guidance mode.

A.5. Scalability and Guidance Pruning

Our experiments evaluate guidance selection on a fixed library. In continual or lifelong settings, however, the guidance library may grow over time, making mechanisms for maintaining scalability important. While we do not empirically study dynamic library growth in this work, we outline a Bayesian-inspired formulation for pruning guidance items that appear unlikely to provide benefit over the baseline model.

Let p_b denote the empirical success probability of the baseline model (i.e., without guidance) on queries that are semantically relevant to a guidance item g. For comparability, both p_b and the success rate of g are estimated over the same relevance-conditioned query subset.

Suppose the posterior over the success probability θ_g of guidance item g, obtained under a uniform prior, is

$$\theta_g \mid \mathcal{D}_g \sim \mathrm{Beta}(\alpha_g, \beta_g),$$

where $\alpha_g - 1$ and $\beta_g - 1$ correspond to the observed numbers of successes and failures.

To assess improvement relative to baseline performance, we form a *baseline-referenced posterior* by combining the observed evidence for g with $\kappa = 2$ baseline-consistent pseudo-observations (success probability p_b). Algebraically, this is equivalent to updating a $\mathrm{Beta}(\kappa p_b,\ \kappa(1 - p_b))$ prior with the observed successes and failures, yielding

$$\tilde{\alpha}_g = (\alpha_g - 1) + \kappa p_b, \qquad \tilde{\beta}_g = (\beta_g - 1) + \kappa(1 - p_b). \tag{4}$$

Let $F_{\mathrm{Beta}(a,b)}$ denote the cumulative distribution function of a $\mathrm{Beta}(a, b)$ distribution. A pruning rule can then be defined by comparing an upper posterior credible bound to the baseline:

$$F^{-1}_{\mathrm{Beta}(\tilde{\alpha}_g, \tilde{\beta}_g)}(1 - \delta) < p_b + \epsilon, \tag{5}$$

where $\delta \in (0, 1)$ specifies the confidence level and $\epsilon > 0$ defines the minimum improvement over baseline considered practically meaningful.

This formulation treats the baseline model as a reference arm. Guidance items are retained only if they are likely, under posterior uncertainty, to provide nontrivial gains over baseline performance. Because κ is small ($\kappa = 2$), the pseudo-observations act only as a weak anchor toward baseline performance. Consequently, items with limited observed data still exhibit relatively high posterior uncertainty, which yields larger upper credible bounds and reduces the risk of premature pruning. As more evidence accumulates, posterior uncertainty shrinks, and items that consistently fail to demonstrate improvement over baseline are progressively removed, helping control library growth.

This perspective is conceptually related to Bayes-UCB methods for multi-armed bandits, which also rely on posterior quantiles for uncertainty-aware decision-making (Kaufmann et al., 2012). A full empirical study of guidance-library dynamics and pruning strategies in large-scale lifelong settings is left for future work.