# TEST-TIME ADAPTATION VIA MANY-SHOT PROMPTING: BENEFITS, LIMITS, AND PITFALLS

**Shubhangi Upasani[1*], Chen Wu[1], Jay Rainton[1], Bo Li[1], Urmish Thakker[1],**
Changran Hu [2], Qizheng Zhang [3]
[1]SambaNova Systems, Inc [2]xAI [3]Stanford University

## ABSTRACT

Test-time adaptation enables large language models (LLMs) to modify their behavior at inference without updating model parameters. A common approach is many-shot prompting, where large numbers of in-context learning (ICL) examples are injected as an input-space test-time update. Although performance can improve as more demonstrations are added, the reliability and limits of this update mechanism remain poorly understood, particularly for open-source models.

We present an empirical study of many-shot prompting across tasks and model backbones, analyzing how performance varies with update magnitude, example ordering, and selection policy. We further study Dynamic and Reinforced ICL as alternative test-time update strategies that control which information is injected and how it constrains model behavior. We find that many-shot prompting is effective for structured tasks where demonstrations provide high information gain, but is highly sensitive to selection strategy and often shows limited benefits for open-ended generation tasks. Overall, we characterize the practical limits of prompt-based test-time adaptation and outline when input-space updates are beneficial versus harmful.

## 1 INTRODUCTION

Test-time adaptation enables large language models (LLMs) to modify their behavior during inference without a change in model weights (Brown et al., 2020; Garg et al., 2022; Zhang et al., 2025). A widely used but underexplored form of such adaptation is prompt-based input-space updates, where task-relevant information is injected directly into the model's context. With recent advances in long-context architectures (Dao et al., 2022; Su et al., 2024), this update mechanism has evolved from few-shot prompting to many-shot prompting, where hundreds or thousands of demonstrations can be provided at test time.

Many-shot prompting can be viewed as an increasingly aggressive test-time update: as more demonstrations are added, the model is exposed to a stronger, task-specific behavioral constraint. Prior work (Agarwal et al., 2024) shows that accuracy often improves as the number of in-context examples grows, particularly for structured tasks. In this paper, we study many-shot prompting as a controlled study on LLaMA model family, focusing on when and why input-space updates succeed or fail. Our key observations from our chosen testbed include the following:

- Many-shot prompting can act as an effective test-time update for structured tasks with accuracy improving consistently up to a moderate update magnitude before saturating.
- Reliability of prompt-based test-time updates is highly sensitive to ordering of in-context demonstrations, prompt structure and example selection strategies.
- Test-time updates exhibit clear task-dependent success and failure modes. Although additional examples provide useful adaptation signals for classification and information extraction tasks, they give small improvements for open-ended generation tasks such as machine translation.
- Structured updates such as Reinforced ICL exhibit similar saturation behavior as update magnitude increases.

*Corresponding author: shubhangi.upasani@gmail.com

Together, these findings highlight both the promise and the limits of prompt-based test-time adaptation.

## 2 BACKGROUND

**Many-Shot Prompting (Update Magnitude):** With the availability of long-context models, in-context learning (ICL) (Dong et al., 2024) has extended from a few demonstrations to many-shot prompting, where hundreds or thousands of examples can be included at test time. The effectiveness of such updates strongly depends on how the context is constructed; demonstration ordering, diversity, template format and phrasing of instructions play an important role.

**Dynamic ICL as Update Policy:** Dynamic ICL (Rubin et al., 2022; Liu et al., 2022) constructs prompts on-the-fly by selecting examples for each query based on similarity, heuristics, or task metadata. It is an adaptive update policy that determines the information that is injected into the context.

**Reinforced ICL as Update Structure:** Reinforced ICL replaces the input-output (IO) demonstrations with reasoning traces, such as chain-of-thought (CoT) examples (Wei et al., 2022; Agarwal et al., 2024), to guide the behavior of the model at the test time. We treat Reinforced ICL as a structured test-time update.

Figure 1 summarizes our view of prompt-based adaptation as a test-time update governed by update magnitude, update policy, and update structure, whose interplay determines whether additional context provides useful signal or introduces noise. Throughout this work, we use **"test-time update"** to refer to input-space adaptation via prompting, without implying any parameter-level modification. We give additional background on the above strategies in Appendix A.1.

## 3 EXPERIMENTAL SETUP

**Tasks and benchmarks:** We evaluate prompt-based test-time updates across a diverse set of tasks using benchmarks designed to stress long-context inference. For structured classification, we use the Banking77 dataset from LongICLBench (Li et al., 2024) which has 77 classes for intent classification task. This dataset enables controlled evaluation of many-shot prompting under extreme label spaces. To assess success and failure modes, we additionally include tasks from Evaluation Harness (Biderman et al., 2024) that span reasoning, information extraction, question answering, and machine translation.

**Models:** We evaluate two instruction-tuned backbones of differing capacity: LLaMA-3.1-8B-Instruct Grattafiori et al. (2024) and LLaMA-3.3-70B-Instruct Meta AI (2024) to study how test-time updates scale with model size under identical prompting strategies. Instruction tuning ensures consistent interpretation of task instructions and demonstrations during many-shot adaptation. Base-model ablations, reported in Appendix A.4, perform poorly even at high update magnitudes, supporting our focus on instruction-tuned models.

## 4 RESULTS: BENEFITS AND LIMITS OF TEST-TIME UPDATES

### 4.1 MORE CONTEXT HELPS—UNTIL IT DOESN'T

We evaluate how the effectiveness of prompt-based test-time updates varies with update magnitude, defined by the total number of in-context demonstrations. For Banking77 with $C = 77$ classes, we use a *per-class* shot formulation: each prompt contains $n$ demonstrations per class, resulting in $N = n \times C$ total demonstrations (e.g., $n = 1 \Rightarrow N = 77$, $n = 5 \Rightarrow N = 385$).

Unless otherwise stated, we refer to $n$ as the *per-class shot count* and $N$ as the *total number of demonstrations*. As shown in Fig. 2(a), as the update magnitude increases for LLaMA-3.1-8B-Instruct, accuracy improves steadily, after which it reaches a plateau at approximately 50–70 shots per class. Beyond this point, additional demonstrations provide diminishing returns. This behavior

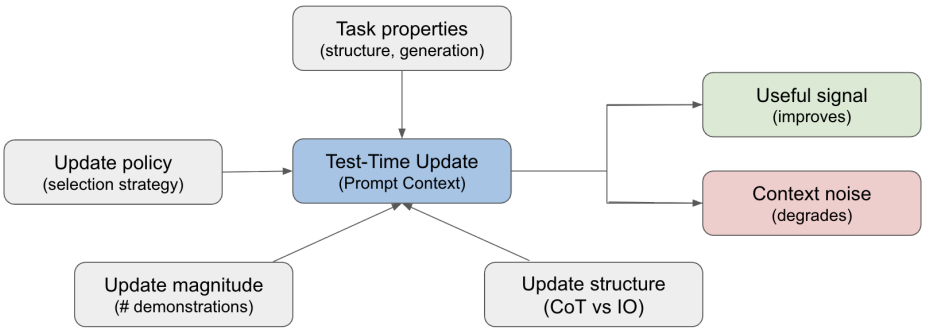


Figure 1: **A unified view of prompt-based test-time adaptation.** Update design determines whether added context provides signal or noise.

indicates a clear saturation regime: moderate input-space updates introduce useful task-specific signal, while aggressive updates yield diminishing further gains Agarwal et al. (2024). We hypothesize that the saturation observed in many-shot prompting arises from the limited representational capacity of attention-based conditioning. Demonstrations influence predictions through their key–value representations in the context, but the resulting update is a weighted aggregation. Once sufficient demonstrations are present to identify the task pattern, additional examples contribute largely redundant representations, causing marginal gains to diminish and, in very long prompts, attention competition can even lead to performance degradation.

To assess update reliability, we repeat each n-shot setting using 10 random demonstration orderings. Fig. 2(a) shows that accuracy varies by 2-3 % across shuffles, demonstrating that many-shot prompting remains sensitive to demonstration ordering due to positional and contextual bias. Averaging over multiple random orders yields more stable results. We explain more experimental details in Appendix A.2.

### 4.2 Update policy matters—relevance helps early, diversity helps at scale

Next, we study how update policy affects test-time adaptation under a fixed context budget in Fig. 2(b) for LLaMA-3.1-8B-Instruct. Across all comparisons, we fix the total number of demonstrations $N$ and vary only the update policy used to select these. We ensure prompt template consistency and prevent including query or same-text duplicates in selected examples.

**Label-wise vs. Cross-label:** We construct prompts using a per-class formulation with $n$ demonstrations per class, resulting in $N = n \times C$ total demonstrations for Banking77. In label-wise selection, we enforce balanced coverage by selecting exactly $n$ demonstrations per class. In cross-label selection, we select $N$ demonstrations from the entire dataset, so label frequencies can be uneven.

**Random vs. similarity:** Within either grouping, random selection samples demonstrations uniformly across the entire dataset, while similarity selection retrieves nearest neighbors to the query in embedding space. More details given in Appendix A.3 and A.5.

Comparing these policies, cross-label selection consistently outperforms label-wise selection, suggesting that enforcing per-label balance can reduce useful diversity by over representing redundant examples. Cross-label benefits from exposure to greater contextual diversity thereby improving generalization, label-balance forces many low information examples which is not very helpful. Cross-label similarity-based selection is strong at small update magnitudes (high relevance), but degrades as $N$ grows, whereas cross-label random selection scales more robustly with larger $N$ (higher diversity). This reflects a bias-variance tradeoff in test-time updates: relevance-focused (similarity-based) policies help early but can over concentrate the context as the update becomes aggressive; diversity-focused (random) policies scale better. Our best setting is ($n = 1 \Rightarrow N = 77$) for cross label similarity. This maximizes relevance per demonstration per class label, yielding strong task adaptation before redundancy degrades performance at larger scale. Since cross-label sampling does not enforce uniform class coverage, improvements in overall accuracy may partially reflect gains on

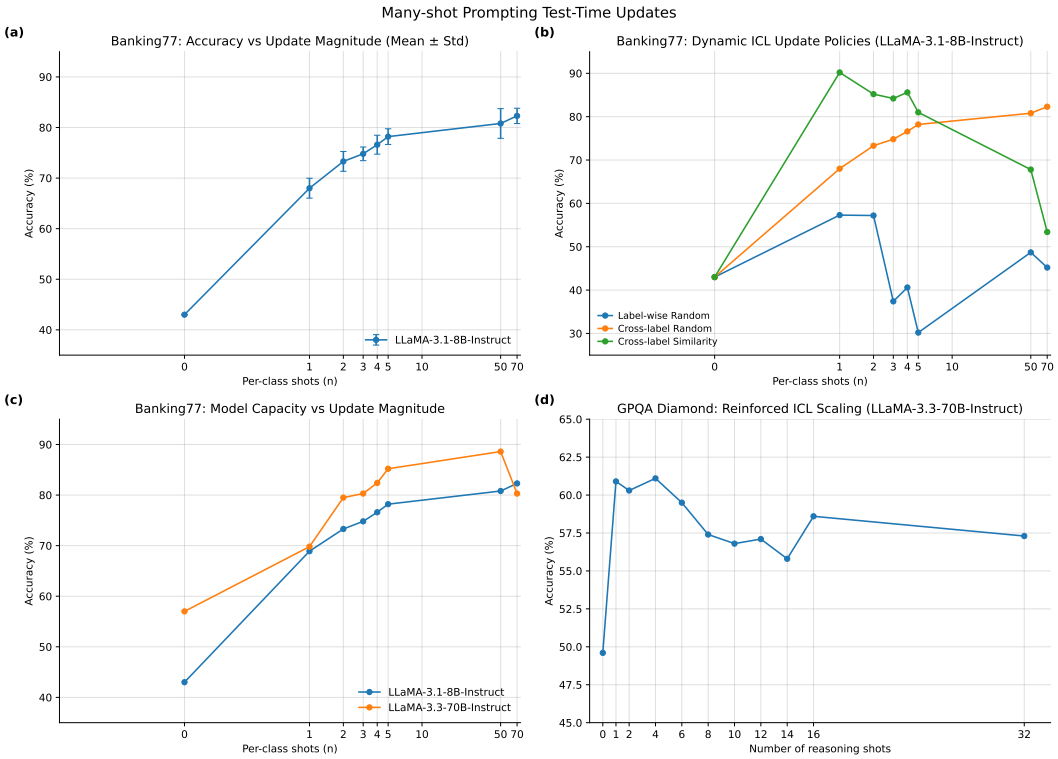


Figure 2: Scaling behavior of prompt-based test-time updates. (a) Many-shot accuracy vs. update magnitude (b) Dynamic ICL update policies (c) Model capacity effects (d) Reinforced ICL scaling

frequent classes; future work could analyze per-class accuracy or macro-F1 to better understand the impact on tail classes. Additional results included in Appendix A.3.

### 4.3 Larger models benefit earlier, smaller models catch up

We further compare test-time update behavior across model sizes using LLaMA-3.1-8B-Instruct and LLaMA-3.3-70B-Instruct in Fig. 2(c) under identical settings for cross label random selection on Banking 77. At small to moderate update magnitudes, LLaMA-70B consistently outperforms the smaller model, indicating that higher capacity models can more effectively exploit diverse in-context supervision. As the update magnitude increases further, the performance gap narrows and the 8B model catches up, suggesting that sufficiently large prompt can partially compensate for limited model capacity. Notably, the 70B model exhibits a performance drop at the largest update magnitude, consistent with over-conditioning. In contrast, the smaller model remains in a signal-accumulation regime and is less sensitive to this effect.

### 4.4 Reinforced ICL exhibits early gains and rapid saturation

We next study Reinforced ICL as a form of structured test-time update, where chain-of-thought (CoT) based reasoning traces Agarwal et al. (2024); Wei et al. (2022) are provided as demonstrations instead of direct input-output pairs. We delineate the data generation pipeline to obtain these samples in the Appendix A.6. Using GPQA Diamond Rein et al. (2024) subset with LLaMA-3.3-70B-Instruct, we show in Fig. 2(d), Reinforced ICL improves performance as the magnitude of the update increases to 4 demonstrations. Beyond this point, accuracy plateaus and degrades. A plausible explanation is that early demonstrations provide a strong inductive bias, leading to rapid gains with only a few examples. As the number of reasoning traces increases, attention is increasingly divided across long chains-of-thought. This competition for attention reduces the effective influence of any single trace, causing performance to plateau or degrade despite additional context Liu et al. (2024).

### 4.5 TASK STRUCTURE DETERMINES THE EFFECTIVENESS OF TEST-TIME UPDATES

Finally, we analyze when prompt-based test-time updates are effective across tasks (Table 5 in Appendix A.7) using LLaMA-3.3-70B-Instruct. We find that many-shot prompting consistently improves performance on structure-heavy tasks with constrained outputs, including structured reasoning (e.g., DROP Dua et al. (2019)) and information extraction benchmarks (e.g., FDA, SWDE Hao et al. (2011)). In these settings, additional demonstrations provide high information gain by capturing relevant patterns. For tasks with constrained outputs (e.g., ARC-Challenge Clark et al. (2018) and GSM8K Cobbe et al. (2021)), performance improves sharply with a small number of demonstrations but quickly saturates, indicating that only limited contextual supervision is required to specify task behavior. In contrast, GPQA (multiple choice) exhibits only modest gains at small update magnitudes. Finally, open-ended generation tasks such as machine translation (WMT16 Bojar et al. (2016)) show consistent but small improvements with additional context, indicating test-time updates offer limited benefits when task structures are already well captured during pretraining.

## 5 CONCLUSION

Our study highlights both the promise and the practical limits of many-shot prompting as a form of test-time adaptation. We show that increasing the number of in-context demonstrations can serve as an effective input-space update for structured tasks. Across tasks and models, performance is highly sensitive to update configuration, including demonstration ordering and selection policy. Dynamic ICL with cross-label selection produces more robust adaptation than static or label-wise prompting, underscoring the importance of update policy design at test time. At the same time, scaling benefits saturate beyond moderate update magnitudes. Finally, we find that many-shot prompting offers limited benefit for open-ended generation tasks, but works well for structured tasks. Taken together, our results suggest that prompt-based test-time updates are most effective when demonstrations inject novel, task-specific information, and that careful control over update magnitude, structure, and policy is essential for reliable deployment. While our experiments focus on a single benchmark, the observed behaviors should be interpreted as preliminary empirical findings, and future work should validate whether these trends generalize across additional datasets and task domains.

## REFERENCES

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966, 2024.

Camel AI. Data generation module — camel ai documentation. `https://docs.camel-ai.org/key_modules/datagen`, 2025. Accessed: 2026-01-XX.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Lucia Specia, Marco Turchi, and Duško Vitas. Findings of the 2016 conference on machine translation (WMT16). In *Proceedings of the First Conference on Machine Translation (WMT16)*, pp. 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `https://aclanthology.org`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. A survey on in-context learning. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 1107–1128, 2024.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in neural information processing systems*, 35:30583–30598, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ming Hao, Jyotish Madhavan, Pedro Szekely, and Alon Y. Halevy. Automatic web data extraction from large websites. In *Proceedings of the VLDB Endowment*, volume 4, pp. 1326–1337. VLDB, 2011.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, William B Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for gpt-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd workshop on knowledge extraction and integration for deep learning architectures*, pp. 100–114, 2022.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Meta AI. LLaMA-3.3-70B-Instruct model card, 2024. URL `https://github.com/meta-llama/llama-models/blob/main/models/llama3_3/MODEL_CARD.md`. Accessed: 2026-01-XX.

OpenAI. Gpt-oss-120b model documentation, 2024. URL `https://platform.openai.com/docs/models/gpt-oss-120b`. Accessed: 2026-01-XX.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2655–2671, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*, 2025.

Table 1: Comparison of per-class shots with total demonstrations in input prompt for Banking 77

| $n$ **per-class shots** | $N$ **Total demonstrations** |
|---|---|
| 1 | 77 |
| 2 | 154 |
| 3 | 231 |
| 4 | 308 |
| 5 | 385 |
| 50 | 3850 |
| 70 | 5390 |

# A APPENDIX

## A.1 ADDITIONAL BACKGROUND ON PROMPT-BASED TEST-TIME ADAPTATION

We provide additional context on how prompt-based adaptation mechanisms differ in the strength, structure, and selectivity of the update signal they introduce at test time.

**In-context learning as test-time updates:** In-context learning (ICL) refers to the ability of large language models to adapt their behavior at inference by conditioning on task-specific examples provided in the input, without modifying model parameters. From a test-time perspective, ICL constitutes an input-space update in which task information is injected directly into the context, enabling adaptation without training.

**Update magnitude in many-shot prompting:** Many-shot prompting increases the strength of test-time update by injecting a larger volume of task-specific signal into the context. While additional demonstrations can improve performance by reinforcing task patterns, they also increase redundancy and noise, making update effectiveness sensitive to prompt composition. In practice, ordering, diversity, and formatting choices determine whether additional examples provide new information or overwhelm the model with repeated context.

**Selectivity in Dynamic ICL:** Dynamic ICL controls which information is injected into the prompt under a fixed context budget. By selecting demonstrations per query, dynamic policies trade off relevance and diversity. Similarity-based selection prioritizes relevance to the input, while broader sampling increases coverage of the task distribution. This selectivity distinguishes dynamic ICL from static prompting, where the same update is applied uniformly across inputs.

**Structure in Reinforced ICL:** Reinforced ICL constrains test-time adaptation through structured reasoning traces rather than direct input–output mappings. By exposing intermediate reasoning steps, the update signal emphasizes how to solve a task instead of what answer to produce. As a result, performance depends not only on the number of demonstrations but also on the consistency and quality of the reasoning patterns being demonstrated.

## A.2 ADDITIONAL EXPERIMENTAL DETAILS FOR UPDATE MAGNITUDE EXPERIMENTS

For the results shown in Fig. 2(a) on Banking77, we vary the per-class shot count $n$. The total number of demonstrations is given by $N = n \times C$, where $C = 77$ is the number of classes. Table 1 illustrates how $N$ scales with $n$. We stop at $N = 5390$ demonstrations, corresponding to the maximum supported context length of 128k tokens for LLaMA models.

To assess update stability, we repeat each setting across multiple random seeds, varying both demonstration ordering and selection, and report mean accuracy along with standard deviation.

## A.3 DYNAMIC ICL SELECTION STRATEGIES

In Dynamic ICL, demonstrations are selected adaptively for each input query rather than using a fixed prompt. We instantiate Dynamic ICL by combining two axes: (i) grouping constraint and

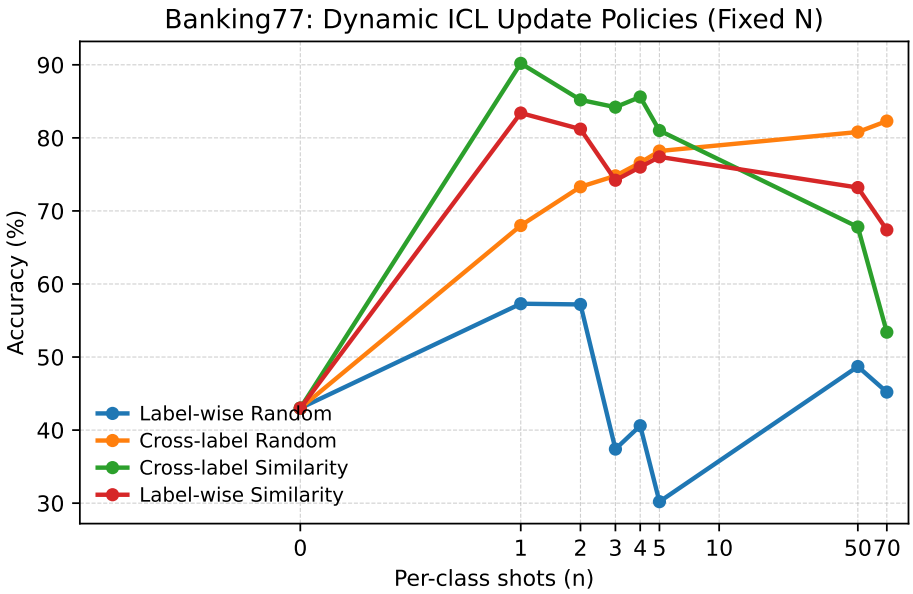


Figure 3: Dynamic ICL Selection Strategies on Banking 77

(ii) retrieval rule. For a dataset with $C$ classes and per-class shot parameter $n$, the prompt contains $N = n \times C$ demonstrations.

**Grouping constraint:**

- **Label-wise:** enforce exactly $n$ demonstrations from each class ($N = nC$).
- **Cross-label:** select $N$ demonstrations globally; class counts are unconstrained.

**Retrieval rule:**

- **Random:** uniform sampling from the entire dataset. The results of random selection are averaged across multiple seeds to ensure consistency.
- **Similarity:** top-$N$ retrieval by embedding similarity to the query.

Combining the above two regimes, we get four strategies used in our experiments:

- Cross-label Random Selection
- Label-wise Random Selection
- Cross-label Similarity Selection
- Label-wise Similarity Selection

This taxonomy allows us to isolate the effects of relevance, diversity, and label balance on test-time adaptation behavior. We make sure that the retrieved examples do not have the query and same-text duplicates, to prevent leakage while evaluating the above selection strategies. We also ensure to keep the instruction and prompt templates constant across all runs. We add label-wise similarity as an additional comparison to Fig. 2(d) and show the full set of results in Fig. 3 for LLaMA-3.1-8B-Instruct.

We observe that label-wise similarity selection consistently outperforms label-wise random selection across all update magnitudes. This indicates that when label balance is enforced, selecting demonstrations that are semantically relevant to the query provides a stronger adaptation signal than random sampling within each class.

However, performance under label-wise similarity selection still degrades as the number of demonstrations increases (e.g., from 83.4% at $n$ = 1 per-class shot to 67.4% at $n$ = 70 per-class shot).

Table 2: Many-shot prompting for Llama-3.1-8B on Banking 77 (accuracy in %). Prompts contain $N$ total demonstrations, constructed using $n$ shots per class ($N = n \times C$, $C$=77).

| Per-class shots $n$ | LLaMA-3.1-8B | LLaMA-3.1-8B-Instruct |
|---|---|---|
| 50 | 33.9 | 80.8 |
| 70 | 35.9 | 82.3 |

This suggests that enforcing per-label balance limits the effective diversity of the prompt: as the update becomes more aggressive, additional demonstrations become increasingly redundant, even when selected by similarity. Label-wise similarity outperforms cross-label similarity at higher n-shots because restricting retrieval to a fixed label avoids conflicting supervision: although examples become increasingly redundant, they remain label-consistent as update magnitude increases. Label-wise similarity, however, underperforms cross-label random selection at larger update magnitudes because the label-wise constraint still limits global diversity and leads to redundancy as more demonstrations are added.

Overall, these results show that while similarity-based selection mitigates some weaknesses of label-wise prompting, the label-wise constraint itself restricts scalability, motivating cross-label selection strategies that allow diversity to grow with update magnitude.

### A.4 Manyshot prompting with Llama-3.1-8B

We compare the LLaMA-3.1-8B base model with its instruction-tuned counterpart under identical many-shot prompting conditions on Banking77, using cross-label random selection. As shown in Table 2, the instruction-tuned model substantially outperforms the base model at both update magnitudes, highlighting the importance of alignment for interpreting task instructions and in-context demonstrations.

At low update magnitudes ($n = 1$–$5$; not shown), the base model performs near chance, indicating a limited ability to infer task structure from few demonstrations. However, as the number of in-context examples increases, the base model exhibits some gains from $n = 50$ to $n = 70$ per-class shot, suggesting that many-shot prompting can induce meaningful test-time adaptation even in the absence of instruction tuning. This result indicates that while alignment strongly improves sample efficiency, the benefits of large-scale in-context updates are not exclusive to instruction-tuned models.

### A.5 Similarity-based retrieval for Dynamic ICL

For similarity based retrieval, we encode all candidate demonstrations and queries using the sentence-transformers/all-MiniLM-L6-v2 model (384-dimensional embeddings). After generating embeddings, we find n-shot nearest neighbors per class to the query embedding for label-wise similairty selection. We make sure that the retrieved examples don't have the query or same-text duplicates. This ensures there is no leakage. Across all retrieval strategies, we use a fixed prompt template; only the selected demonstrations vary between conditions.

### A.6 Data generation recipe for Reinforced ICL

To generate synthetic reasoning traces for Reinforced ICL on GPQA Diamond shown in Fig. 2(d), we follow the Chain-of-Thought data generation pipeline described in the Camel AI framework AI (2025). We instantiate a generator–verifier setup, where both agents are powered by the gpt-oss-120b OpenAI (2024) model. The generator produces step-by-step reasoning traces under predefined formatting constraints, and the verifier checks the final answer against the ground-truth label. We retain only examples whose generated final answer exactly matches the correct label, filtering out inconsistent traces. Applying this procedure to the 198 question in GPQA Diamond yields 133 validated reasoning demonstrations, which are used for Reinforced ICL experiments. We used greedy sampling with max tokens set to 12000 to generate CoT traces. We have shown the generator and verifier prompt in Table 3 and Table 4 respectively.

Table 3: Prompt template used for the Generator agent during data generation for Reinforced ICL.

| **Generator Prompt** |
|---|
| Generator agent for creating step-by-step reasoning. Rules:<br>– Use at most 10 steps.<br>– No explanations.<br>– No restating the question.<br>– End immediately after the answer.<br>– Do not reference the order of the answers in the question (e.g., option a, b, c).<br>– For the answer in the <a> tags, ensure it exactly matches one of the listed options.<br><br>Format:<br><r><br>1. ...<br>2. ...<br></r><br><a><br>...<br></a> |

Table 4: Prompt template used for the Verifier agent during data generation for Reinforced ICL.

| **Verifier Prompt** |
|---|
| Verified agent for step by step reasoning.<br>For the text within the <a> tags, make sure it matches exactly to the correct answer, word for word. |

### A.7 Many-shot prompting on Evaluation Harness

We report full results across Evaluation Harness tasks with LLaMA-3.3-70B-Instruct in Table 5 to characterize where many-shot test-time updates succeed and where they fail. These experiments span structured reasoning, and open-ended generation settings, allowing us to examine how the effectiveness of prompt-based updates depends on task structure and information content. The complete results provide finer-grained evidence for the task-dependent patterns discussed in the main paper, and highlight regimes where additional in-context demonstrations introduce useful adaptation signals versus redundant or noisy context. We use BLEU score to evaluate machine translation (WMT16) tasks, F1 score to evaluate reading comprehension (DROP) task and accuracy (in %) based on exact match for the rest of the tasks.

Table 5: Task-dependent performance of many-shot test-time updates (LLaMA-3.3-70B-Instruct)

| Task | Baseline | 4-Shot | 16-Shot | 32-Shot |
|---|---|---|---|---|
| **Domain-Specific / Information-Heavy Tasks** | | | | |
| DROP | 10.8 | 13.1 | 14.2 | 15.8 |
| FDA | 39.65 | 86.8 | 89.4 | 89.7 |
| SWDE | 74.17 | 92.9 | 95.1 | 96.4 |
| **Structured Reasoning Tasks (Constrained Output)** | | | | |
| ARC-Challenge | 38.82 | 93.72 | 93.48 | 93.45 |
| GSM8K | 87.56 | 94.7 | 94.8 | 94.8 |
| GPQA (MC) | 47.99 | 51.9 | 50.0 | 48.8 |
| **Open-Ended Generation Tasks** | | | | |
| WMT16 En-De | 34.51 | 37.0 | 37.4 | 37.5 |
| WMT16 De-En | 44.76 | 46.3 | 46.9 | 47.0 |

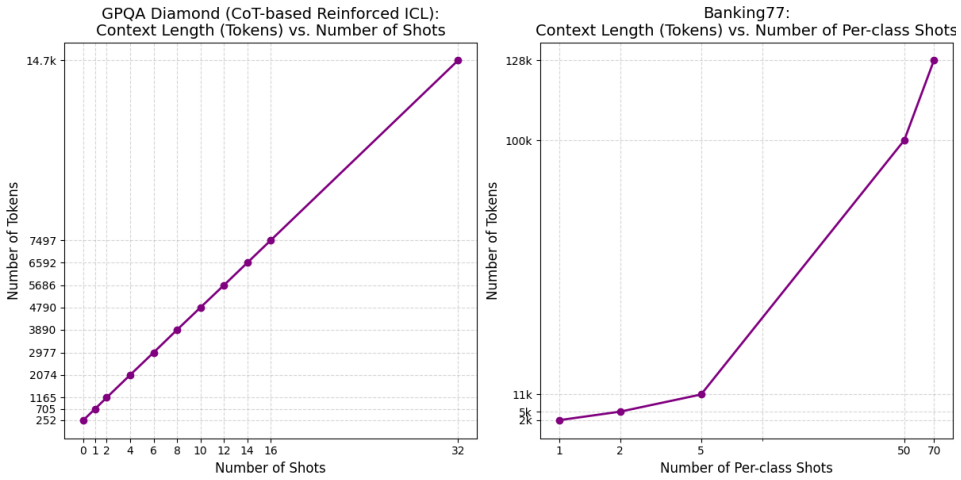


Figure 4: Context length scaling with update magnitude

### A.8 Context growth under many-shot and reinforced test-time updates

Figure 4 illustrates how prompt context length (in tokens) scales with the number of demonstrations for two settings: Reinforced ICL on the GPQA Diamond subset (left) and many-shot prompting on Banking77 (right). For Reinforced ICL, the context length grows approximately linearly with the number of reasoning demonstrations included in the prompt, reaching a maximum input length of 14.7k tokens at 32-shot. For Banking77, we vary the number of per-class shots $n$; the total number of demonstrations in the prompt is $N = n \times C$, where $C = 77$ is the number of classes. At $n$=70 ($N$=5390), the prompt reaches the maximum supported context length of 128k tokens.