Where to show Demos in Your Prompt:
A Positional Bias of In-Context Learning

Anonymous ACL submission

Abstract

In-context learning (ICL) is a critical emerging capability of large language models (LLMs), enabling few-shot learning during inference by including a few demonstrations (demos) in the prompt. However, it has been found that ICL's performance can be sensitive to the choices of demos and their order. This paper investigates an unexplored new positional bias of ICL for the first time: we observe that the predictions and accuracy can drift drastically when the positions of demos, system prompt, and user message in LLM input are varied. This bias, we refer to as DEMOS' POSITION IN PROMPT bias (DPP bias). We design a systematic evaluation pipeline to study this type of positional bias across classification, QA, summarization, and reasoning tasks. We introduce two metrics, ACCURACY-CHANGE and PREDICTION-CHANGE, to quantify net gains and output volatility induced by demos' position change. Extensive experiments on ten LLMs from four open-source model families (QWEN, LLAMA3, MISTRAL, COHERE) verify that the bias significantly affects their accuracy and predictions: placing demos at the start of prompt yields the most stable and accurate outputs with gains of up to +6 points. In contrast, placing demos at the end of the user message flips over 30% of predictions without improving correctness in QA tasks. Smaller models are most affected by this sensitivity, though even large models do remain marginally affected on complex tasks.

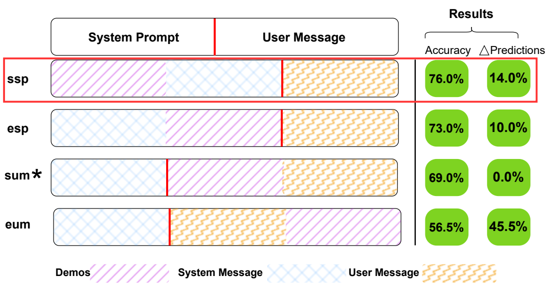


Figure 1: **Four configurations of demos' position in prompt (DPP)** from §3: *ssp* (Start of System Prompt), *esp* (End of System Prompt), *sum* (Start of User Message, **default**), and *eum* (End of User Message). Their results with QWEN-1.5B on AG news datasets are reported on the right: Their accuracies vary drastically and the percentage of changed predictions (compared to default *sum*) can be up to *45.5%*.

1 Introduction

The rapid evolution of large language models (LLMs) has redefined the boundaries of machine learning, enabling unprecedented few-shot and zero-shot generalization across tasks like classification, question answering, and summarization (Brown et al., 2020; Radford et al., 2019). Central to this paradigm shift is in-context learning (ICL), where models dynamically adapt to new tasks by processing demos embedded directly in the input prompt. Recent work has exposed critical vulnerabilities: minor perturbations to demo ordering or demo count (Lu et al., 2022) can degrade performance unpredictably. This brittleness not only undermines reproducibility but also challenges assumptions about LLMs' capacity for systematic reasoning, raising urgent questions about whether current models truly learn from context or merely exploit superficial patterns.

We discover a **novel positional bias** in *in-context learning* (ICL): DPP bias, in which moving an *unchanged* block of demos from the start of a prompt to the end can swing task accuracy by **up to 20 percentage points** and flip almost half of a model's predictions (see Fig. 1). This phenomenon, purely spatial, independent of demo content, challenges the widespread assumption that large language models (LLMs) learn robustly from any properly formatted context.

Despite growing awareness of prompt sensitivity, the role of demo positioning where demos are placed relative to instructions, queries, or other contextual elements remains underexplored. Prior studies have focused primarily on demo selection

(Liu et al., 2022), or template phrasing (Cho et al., 2024; Voronov et al., 2024), leaving a gap in understanding how spatial arrangements modulate ICL efficacy. This paper addresses this gap through a systematic investigation of positional effects across eight tasks spanning classification, reasoning, and generation. By conducting controlled studies on models like LLAMA-3 (1B, 3B, 8B, 70B) and MIXTRAL_8X7B, we demonstrate that strategic placement (e.g., clustering critical demos near task instructions) can yield performance swings, even when demo content remains identical.

Our work makes **five complementary contributions**. **1.** We first uncover and quantify a previously unreported positional bias (DPP bias) in in-context learning, showing that simply relocating an identical demo block within the prompt can shift accuracy by up to 50 percentage points while flipping nearly half of a model's predictions. **2.** Building on this insight, we design a controlled evaluation pipeline that isolates four canonical demo placements, at the start or end of the system prompt and at the start or end of the user message, so that any performance change is attributable purely to position. **3.** To capture both net performance shifts and output volatility, we introduce two task-agnostic metrics, *accuracy-change* and *prediction-change*. Using this framework, **4.** we conduct the first large-scale empirical study of positional effects across eight tasks and ten state-of-the-art LLMs, revealing a consistent primacy bias that becomes less severe as model size grows. **5.** Finally, we translate these findings into practical guidelines.

2 Related Work

In this section, we review existing literature on positional biases in in-context learning (ICL). We organize the discussion into three subsections: internal demo-order bias, mechanistic hypothesis, and the role level gap spatial placement.

2.1 Internal Demonstration-Order Bias

Prompt-order sensitivity is a well-established phenomenon in in-context learning (ICL). Lu et al. (2022) demonstrated that merely permuting the order of demonstrations can lead to accuracy fluctuations of approximately ±15% in reasoning tasks, such as arithmetic and commonsense question-answering. Similarly, Min et al. (2022) found that large language models (LLMs) frequently exploit superficial lexical overlaps between demonstrations and queries rather than learning robust semantic mappings. Zhao et al. (2021) further showed that demonstration order significantly impacts few-shot outcomes and this was also supported by Wang et al. (2023) who found that ChatGPT predominantly favors earlier listed labels in classification tasks, while Wei et al. (2022b) indicated that reasoning gains from Chain-of-Thought (CoT) rationales heavily depend on their positioning within prompts. These studies underscore the fragility of ICL to superficial prompt characteristics, motivating further exploration into position-related biases. *Our study departs from these works by holding the* internal *order fixed and relocating the entire demo block to different prompt regions.*

2.2 Mechanistic Hypothesis

Recent research attributes positional bias in transformer-based models to intrinsic architectural tendencies, notably primacy bias and induction heads. Olsson et al. (2022) and Chan et al. (2022) highlight that transformers disproportionately emphasize early tokens due to induction head mechanisms, causing initial context to steer subsequent predictions significantly. Similarly, Xiao et al. (2024) note sequential processing biases towards earlier context, which impact performance when crucial information appears later in the sequence. Additionally, Liu et al. (2023) observed that tokens in the middle positions of sequences receive less attention, leading to performance degradation. Bietti et al. (2023) further supports this by linking primacy bias to underlying transformer memory mechanisms. While these hypotheses illuminate *why* order matters, empirical work on how they interact with *prompt roles* (system vs. user) is scarce. *We provide the first role-aware stress test of these mechanisms.*

2.3 Spatial Placement (Role-Level) Gap

While prior ICL research extensively explores the selection of demonstrations, relatively little attention has been paid to their precise spatial placement within prompts. Studies such as Cho et al. (2024), Reynolds and McDonell (2021), and Webson and Pavlick (2022) prioritize choosing semantically relevant demonstrations and designing tailored prompt templates but overlook how the exact location of demonstration blocks, particularly relative to system and user roles, might independently affect model outcomes. Our study addresses this gap by explicitly varying demonstration placement across prompt roles, highlighting an overlooked but critical dimension of prompt structuring for achieving reliable ICL performance.

3 Methodology

We present a systematic framework to investigate how the position of in-context demos within a prompt affects model performance. Our approach formalizes the problem of DPP bias, defines the range of demonstration placements considered, and outlines an evaluation pipeline for measuring performance variations.

3.1 Problem Formulation

We focus on the classical in-context learning scenario, where a large language model (LLM) is given a small set of demonstrations along with a query, all concatenated into a single prompt. Formally, let $\mathcal{T}$ be a set of tasks (e.g. sentiment classification, QA, etc.), and for each task $\tau \in \mathcal{T}$, let D_τ be a set of N demonstrations and Q_τ a set of evaluation queries. For a given query $q \in Q_\tau$, we construct a prompt P that combines some or all examples from D_τ with q. Crucially, our study keeps the content of P (the instruction, the examples in D_τ, and the query q) fixed, and manipulates only the structural position of the demonstration block within the prompt. We define positional bias (or spatial confounder effect) as any change in the model's performance on the query set Q_τ that arises solely from where the demonstrations appear in P, rather than which demonstrations are provided. Essentially isolating how the different structural positions affect the model output.

3.2 Demo Positions: Definitions

In many recent instruction tuned LLMs, a prompt can include a system prompt, which is then followed by the user message (chat-style format). We leverage this structure to define four distinct canonical demonstration positions where a block of k demos can be inserted in the prompt. These four configurations, illustrated in Figure 1 are defined as followed:

- Start of System Prompt (*ssp*): The demos block is placed at the very beginning of the system message, before any instructional content.

- End of System Prompt (*esp*): The demos block is placed at the end of the system message, after any general instructions but still before the user's query.

- Start of User Message (*sum*): The demos block is inserted at the beginning of the user message, before the actual query text.

- End of User Message (*eum*): The demonstration block is appended at the very end of the user message, after the query.

Figure 1 provides a schematic diagram of these four positions. It shows whether the demos reside in the system vs. user section of the prompt and whether they appear at the start or end. Intuitively, *ssp* and *esp* represent placing demonstrations before the user's question, whereas *sum* and *eum* place them before and after the user's question respectively.

3.3 Evaluation Metrics

We report the task-specific metrics recommended by prior work: **Accuracy** for multiple-choice (MCQ) problems, $\mathbf{F}_1$ and **Exact Match** for extractive question answering (QA), and **ROUGE**–L and **BERTScore** for summarization. Aside from the suggested metrics, to understand the per question by position transitions, we also report other metrics:

Accuracy Change. Accuracy Change Δ_{metric} directly quantifies how adding demonstrations at a given position influences the model's overall task performance relative to zero-shot. Formally,

$$\Delta_{\text{\$metric}} = \text{Metric}_{\text{position}} - \text{Metric}_{\text{zero-shot}} \tag{1}$$

A positive $\Delta_{\text{\$metric}}$[1] indicates that placing demos in that location helps the model make more correct predictions, while a negative value means the demonstrations actually degrade performance. By isolating the net gain or loss in accuracy, this metric cleanly attributes performance differences to spatial placement of the same content, enabling fair comparison across positions, models, and tasks.

Prediction Change. Prediction Change Δ_{pred} measures the volatility of individual model outputs induced by demonstration placement. It is defined as

$$\Delta_{\text{pred}} = \frac{\#\textit{answer flips}}{\#\mathcal{Q}} \tag{2}$$

[1] metric = Accuracy, Exact Match, ROUGE-L

where $\#Q$ is the total number of evaluation queries, and *#answer flips* counts all instances whose predicted outputs flips when going from the default ICL position(*sum*) to the other in-context positions. High Δ_{pred} reveals that demonstration placement strongly perturbs the model's decision boundary, even if net metric gains are small.

Remarks We propose a systematic framework to investigate how the structural position of in-context demonstrations affects large language model (LLM) performance. Our study isolates positional effects by controlling for prompt content while varying the location of a fixed demonstration block. We define four canonical positions within a prompt, *ssp*, *esp*, *sum*, and *eum*, which differ in whether demos are placed within the system or user section, and whether they precede or follow the query. These positions are visualized in Figure 1.

4 Empirical Results

We evaluate how demonstration position affects model performance both in terms of net accuracy change relative to zero-shot, and in terms of answer volatility (prediction flips)

4.1 Positional Bias across Tasks

A consistent and pronounced pattern emerges across our benchmark datasets: demonstrations positioned at the beginning of prompts (*ssp* or *esp*) reliably outperform placements later in the prompt (*eum*) and frequently surpass the default ICL position (*sum*). Throughout our experiments, we set the number of demos to five. We keep the demos in the demos block and identical across these conditions, so that any performance differences can be attributed purely to positional effects. (Any additional prompt formatting details and exact templates used for each position are provided in the Appendix. §A.2 & §A.3)

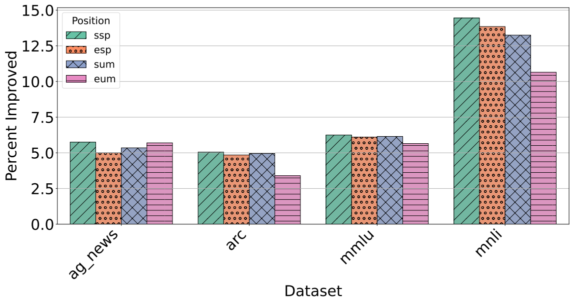


Figure 2: **Accuracy Change (comparing to zero-shot) of the four `DPPs` across four datasets, averaged over all models.** The `ssp` achieves the greatest improvement over zero-shot across all four datasets (note the winner may vary for different models as shown in Fig. 8-10).

Classification and QA Tasks. Across MNLI, AG News, ARC, and MMLU, placing demonstrations at `ssp` yields the most consistent accuracy improvements (Figure 2). Notably, MMLU shows a *+18%* gain in accuracy over the zero-shot baseline under `ssp`. For QA tasks like SQuAD, `ssp` similarly outperforms later placements, while `eum` consistently underperforms.

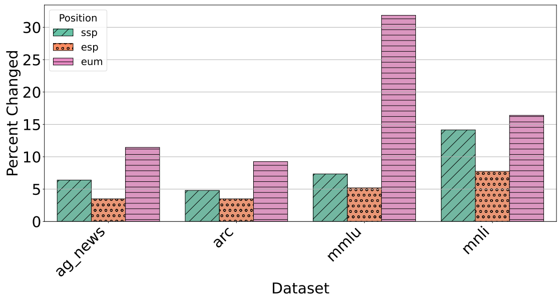


Figure 3: **Prediction change (comparing to `sum`) ratios of the three `DPPs` (excluding `sum`) across four datasets.** The `eum` position shows the largest variability on the `mmlu` dataset.

Arithmetic Tasks. Arithmetic reasoning exhibits scale-sensitive trends. When evaluated, models with smaller parameter sizes (1.5B - 8B) are consistent in preferring demos being placed in the *ssp, esp* positions. For LLAMA3 3B, moving demos from `ssp` to `eum` causes a drop in improved prediction rate: GSM8K falls from *42.0%* to *11%*, and SQuAD from *41.0%* to *26.5%*. Conversely, LLAMA3 70B benefits from `eum`, improving from *21.5%* to *88%* on GSM8K, suggesting that model capacity modulates the effect of position.

Generative Summarization. Performance volatility is most severe in generation tasks. On LLAMA3 3B, XSum sees a drop from *82.5%* to *27.5%* improved predictions when shifting from `ssp` to `eum`, while CNN/DailyMail drops from *49%* to a mere *1%*. These effects percist even in large models, albeit with reduced severity.

4.2 Scaling Law of Performance Robustness

To better understand how positional robustness varies with model scale, we analyze the percentage of changed and improved predictions across the four prompt positions. Across all tasks, we observe that larger models generally exhibit reduced prediction volatility (*% changed*) and enhanced performance stability, but the degree of robustness is task-dependent and not uniformly monotonic with size.

Stability Trends Across Tasks. On **classification tasks** such as *AG News*, *MNLI*, and *ARC*, larger models (e.g., Qwen 72B, LLAMA3 70B) exhibit

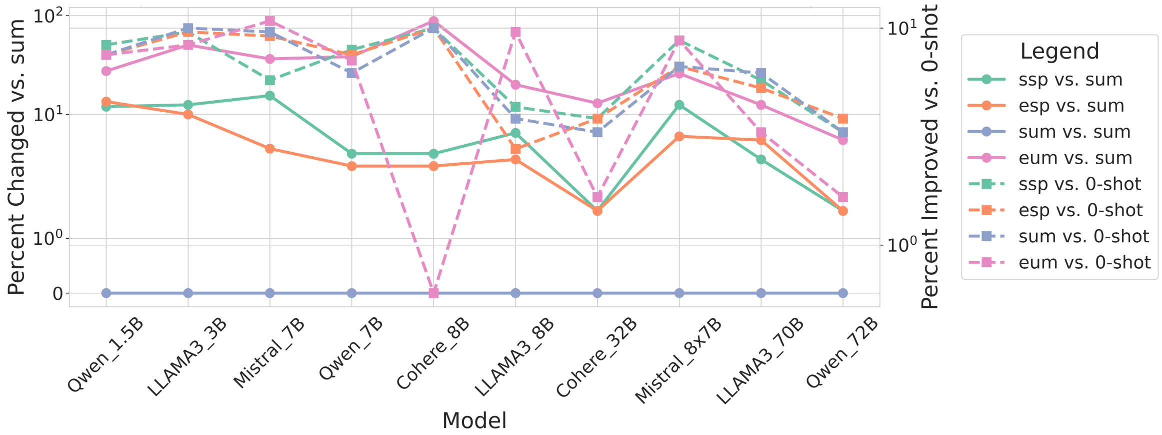


Figure 4: **Prediction change (vs. zero-shot) and accuracy change (vs. sum)** of the four DPPs when applied to **10 LLMs of different sizes (1.5B to 72B)** on MMLU. Both metrics reveal a weak scaling law: as the model scale increases, the variations caused by DPPs in accuracy and prediction from baselines gradually decline.

reduced sensitivity to prompt position changes, especially for early-positioned demonstrations (ssp, esp). For example, on *MNLI*, the percentage of predictions that change when moving from sum to ssp drops below *10%* for LLaMA3 70B, compared to over *20%* for LLaMA3 3B. Meanwhile, accuracy improvements over zero-shot are consistently higher for early positions but show greater spread across mid-sized models (e.g., 7B–32B). This indicates that while small models benefit from positional tuning, they are also more fragile to changes.

On **question answering tasks** like *SQuAD* and *GSM8K*, the pattern is more nuanced. For *GSM8K*, the change rate remains above *90%* across nearly all models and positions, indicating high sensitivity to demonstration placement. However, the percentage improvement fluctuates non-monotonically: models like Mistral 8x7B under-perform with ssp placement relative to both smaller and larger models, and LLaMA3 70B shows a complete collapse in improvement under ssp, contrasting its robustness on other tasks. This suggests arithmetic reasoning requires specialized inductive biases that do not scale uniformly with size.

In **summarization tasks** such as XSum and CNN/DailyMail, the percentage of prediction changes is consistently near *100%* for the eum position, even in the largest models. This reflects that downstream text generation is highly susceptible to positional shifts. Notably, larger models like Qwen-72B still exhibit drops in *% improved* when moving from ssp to eum, albeit less drastically than smaller counterparts. On *CNN/DailyMail*, eum improves only 1% of predictions for LLaMA3-3B, compared to *49%* under ssp, while LLaMA3-70B narrows that gap considerably.

4.3 Analysis of DPP induced Transitions

While accuracy-based evaluations reveal global trends in positional effectiveness, they can obscure local instability in model behavior. To uncover finer-grained effects, we visualize the answer transitions between correct and incorrect predictions using Sankey diagrams.

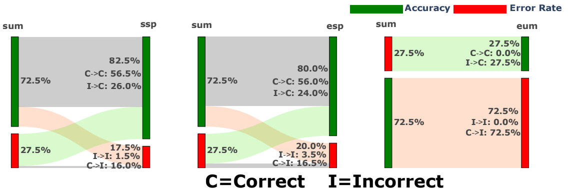


Figure 5: **Correct-Incorrect Transition** from the default baseline DPP "sum" to ssp, esp, and eum when applied to LLaMA3-3B model on *XSUM* benchmark. Green and red bars denote the accuracy and error rate, respectively. Left and right bars are associated with the baseline and a specific DPP. We also report the percentage of examples that change from Incorrect→Correct (**I -> C**) and Correct→Incorrect (**C -> I**).

Volatility Patterns Across Tasks. Across the eight evaluated tasks, we observe a recurring pattern: later-positioned demos (eum) cause significantly more answer flips than earlier positions (ssp, esp). This suggests that placing demonstrations after the query can inject instability into model decision-making, especially in models with fewer inductive biases or weaker context modeling capabilities.

In Figure 5, we see this volatility concretely for LLaMA3 3B on *MMLU*, where moving from ssp to eum causes a large number of transitions from correct to incorrect answers. Similar patterns are seen on:

- ***AG News***: Smaller models like Qwen 1.5B exhibit over 40% incorrect-to-correct transitions under ssp, which plummet under eum.
- ***CNN/DailyMail***: Mistral 8x7B shows one of the most volatile behaviors, with many correct answers flipping to incorrect under late-positioned demos (Fig. 6).
- ***GSM8K***: Predictions by models like Qwen 72B and LLAMA3 70B still flip a lot across positions, despite their scales (Fig. 7).

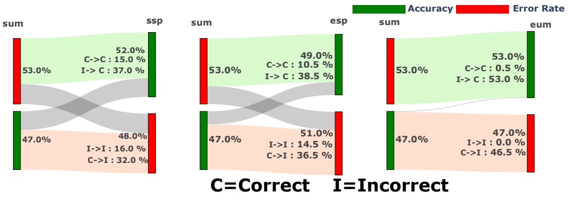


Figure 6: **Correct-Incorrect Transition** on CNN/ DailyMail for Mistral-8x7B. The high transition ratios between incorrect and correct samples indicate the sensitivity to the change of DPP.

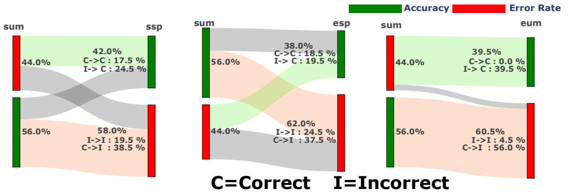


Figure 7: **Correct-Incorrect Transition** on GSM8K for Qwen-72B. Even for the largest model evaluated in this paper, >50% predictions are changed when using different DPP.

Together, these transition plots reveal that *the same input content,when moved across prompt sections can yield drastically different outputs*. The effect persists across models and tasks, underlining that prompt formatting is not merely stylistic, but functionally consequential. This volatility is especially concerning in high-stakes domains like QA or summarization, where reliability is paramount.

Scale-Driven Shifts in Optimal Position. Importantly, the position yielding the best improvement is not consistent across model sizes. On *ARC*, ssp dominates for smaller models (Qwen 1.5B to Mistral 7B), whereas eum unexpectedly overtakes ssp in Qwen 72B albeit marginally. Similarly, on *AG News*, while ssp yields the best result for LLaMA3 3B, esp becomes the strongest position in LLaMA3 70B.

4.4 Winning DPP is Task and Model Specific

While general trends suggest that early demonstration positions (ssp, esp) often outperform later ones (sum, eum), our analysis reveals that this preference is not consistent across all models or tasks. To understand this heterogeneity, we conduct a win–tie–loss analysis across tasks, identifying which demo position performs best for each task–model pair.

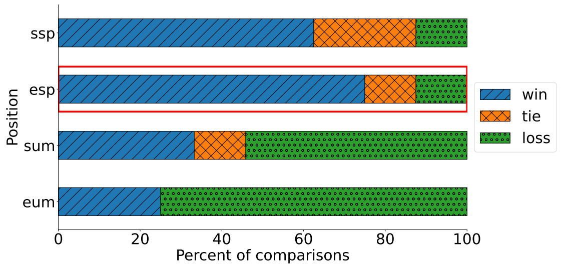


Figure 8: Win–loss–tie of each DPP vs. zero-shot on Qwen 1.5B (averaged over all the eight benchmarks).

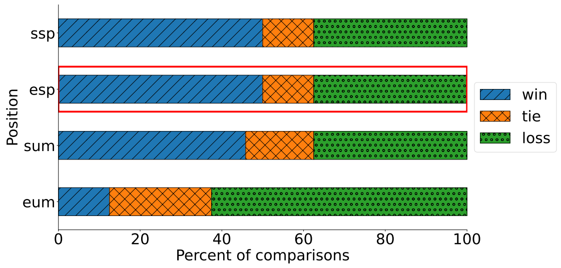


Figure 9: Win–loss–tie of each DPP vs. zero-shot on Cohere 8B (averaged over all the eight benchmarks).

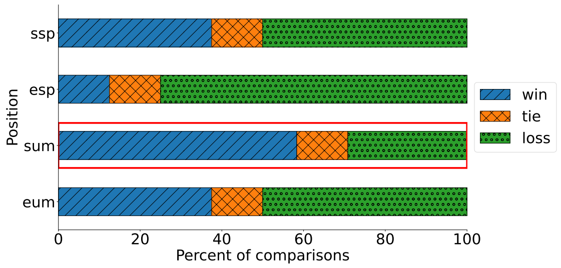


Figure 10: Win–loss–tie of each DPP vs. zero-shot on LLaMA3 70B (averaged over all the eight benchmarks).

Figures 8, 9, and 10 illustrate this breakdown for three representative models at different scales: Qwen-1.5B, Cohere-8B, and LLaMA3-70B. These win–loss–tie plots display, for each position, the number of tasks where it yielded the best performance (win), tied for the best (tie), or was outperformed by another position (loss).

Qwen 1.5B (Figure 8): As the smallest model in our suite, Qwen 1.5B strongly prefers placing demos at the esp and ssp position. It wins on most tasks with esp and ssp, rarely losing. This suggests that smaller models are especially sensitive to how demonstrations are front-loaded in the prompt, likely due to limitations in long-range context integration.

Cohere 8B (Figure 9): At 8B parameters, Cohere shows moderate flexibility. While ssp still wins most often, sum begin to win on some tasks, particularly XSUM and SQUAD indicating that as model capacity grows, preferences start to shift depending on task format and type (classification vs. QA vs. generation).

LLAMA3 70B (Figure 10): In contrast to smaller models, LLAMA3 70B shows a consistent preference for placing demonstrations at the sum position, that is, at the start of the user message. Across multiple tasks, sum outperforms all other configurations, including ssp and esp, which dominate in earlier models. This suggests that larger models like LLAMA3 70B may benefit from having demonstrations placed in closer proximity to the query, perhaps due to their greater ability to retain relevant context across longer input sequences.

Emergent Observation: No Universally Best Position. Our results demonstrate that early positions dominate on average but exceptions emerge for arithmetic tasks Instead, the optimal position varies by both model architecture and task category. For example, in generative summarization tasks, later positions (sum, eum) occasionally outperform early ones, whereas in classification and reasoning tasks, early positions (ssp, esp) are generally more reliable.

For completeness, we provide win–loss–tie plots for all remaining models and also task specific plots in the §A.8 (Figures 11–17). Collectively, they confirm the absence of a universally optimal position and highlight the need for model-specific prompt tuning.

4.5 Statistical Test of Performance Difference between zero-shot vs. ICL with each DPP

Position	0-shot Accuracy	ICL Accuracy	p-value	Effect Size
ssp	0.3364	0.6885	0.0022**	1.7193
esp	0.3364	0.6950	0.0022**	1.7000
sum	0.3364	0.6869	0.0022**	1.7254
eum	0.3364	0.4519	0.1659	0.4140

Table 1: Comparing zero-shot vs. the four DPPs on MMLU dataset (averaged over all models) via one-sided Wilcoxon signed-rank test. **–statistical significance at 1%.

To quantify the reliability of performance differences across demonstration positions, we conduct a paired statistical analysis comparing each of the four DPPs to the zero-shot baseline.

For each dataset and DPP, we form paired samples across the available models. We then perform a one-sided Wilcoxon signed-rank test to assess whether the positional condition of the ICL improves over baseline. Specifically, we test the null hypothesis H_0**: the median difference between the DPP and the baseline is zero**, against the alternative hypothesis H_1**: the median difference is greater than zero**, indicating that the DPP outperforms the baseline. The effect sizes are calculated as the standardized mean difference of paired differences. In addition, we apply a multiple comparisons correction (using the FDR Benjamini–Hochberg procedure at $\alpha = 0.05$) to account for the fact that multiple hypotheses are tested simultaneously. This analysis provides statistical rigor to our evaluation, helping us determine not just whether differences exist, but whether they are consistently positive across models. By quantifying both the statistical significance and effect size, we can better assess the reliability and practical importance of each DPP.

5 Conclusion

This paper introduces and systematically investigates a previously overlooked dimension of in-context learning (ICL): the effects of the positional placement of demonstrations within LLM prompts. Through a large-scale evaluation spanning ten open-source models, eight NLP tasks, and four canonical prompt positions, we uncover a consistent DPP bias, where demos placed earlier in the prompt (ssp, esp) yield higher accuracy and greater prediction stability than those placed later (sum, eum). These findings persist across both classification and generative tasks and are particularly pronounced in smaller models.

Our analysis reveals that not only does performance vary substantially by position, but late-placed demonstrations (especially eum) can induce significant prediction volatility flipping model outputs without improving correctness. We further show that positional sensitivity is modulated by both task and model scale: while larger models demonstrate greater robustness, they still exhibit non-trivial instability and shifting optimal positions across tasks.

We introduce novel diagnostic tools, ACCURACY-CHANGE and PREDICTION-CHANGE to quantify these effects and uncover hidden volatility that standard accuracy metrics obscure. Our win–tie–loss analyses reinforce the key insight: **no single demonstration position is universally optimal**.

Effective prompt design must therefore be both *model-aware* and *task-sensitive*.

These findings have broad implications for prompting strategies in practice. We recommend that users of instruction-tuned LLMs explicitly evaluate demonstration placement rather than relying on default or ad hoc formats. Furthermore, positional robustness should be considered a core axis in both prompt optimization and instruction fine-tuning pipelines.

Future Work. Our study opens up several avenues for follow-up research. First, deeper interpretability work could investigate *why* certain positions are privileged, whether due to attention initialization, decoder primacy, instruction tuning templates or training corpus conventions. Second, extending this analysis to few-shot chain-of-thought prompts and real-world instruction datasets (e.g., HELM, BIG-Bench) could help generalize these insights. Finally, developing automated demo-placement optimization routines that adapt position jointly with content could offer a principled pathway toward more robust ICL systems.

6 Ethics Statement

Our work focuses on the technical aspects of prompt design and does not directly engage with potentially sensitive content or private data. However, the following ethical considerations are relevant:

1. **Misuse of Prompt Engineering:** Enhanced control over LLM behavior through strategic demonstration placement could be exploited to generate deceptive or harmful content more effectively. We encourage researchers to incorporate content filtering and moderation frameworks when deploying these methods.

2. **Bias and Fairness:** If demonstrations carry implicit biases (e.g., skewed label distributions or stereotypical examples), placing them early in the prompt may amplify such biases in model outputs. Practitioners should carefully curate demonstration sets and validate outputs for unintended bias.

We believe that increasing awareness of spatial effects in prompts will ultimately aid in designing safer, more reliable LLM-based systems while mitigating misuse and bias.

7 Limitations

While our experiments reveal robust trends in how demonstration placement impacts LLM performance, several limitations remain:

- **Model Diversity:** We evaluated only a small subset of model sizes and architectures (e.g., 7B, 13B). Larger-scale models or different architectures (e.g., those fine-tuned on dialogue) may exhibit different sensitivity patterns.

- **Task Coverage:** Though we tested multiple tasks (classification, QA, summarization, reasoning), certain tasks with more complex structures (e.g., multi-hop retrieval or dialogic contexts) were not explored in depth.

- **Focus on English:** Our results primarily focus on English data. Cross-lingual variations in grammar, morphology, and script may lead to different positional biases and should be investigated further.

- **Automated Evaluation Metrics:** We relied on standard metrics (accuracy, F1, ROUGE) to quantify performance. These are imperfect proxies for true utility, especially for generative tasks. It's conceivable that a prompt layout yields a higher ROUGE but lower factuality, for example. We assume the metrics correlate with better quality in our tasks, which is generally accepted, but caution that "better metric" doesn't always mean strictly better output in all aspects.

Addressing these limitations will be crucial for fully understanding the impact of demonstration placement across diverse LLMs, languages, and application domains. We hope our findings will catalyze more research into robust, spatially aware prompting techniques.

References

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. 2023. Birth of a transformer: A memory viewpoint. *Preprint*, arXiv:2306.00802.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens

Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Stephanie C. Y. Chan, Adam Santoro, Andrew K. Lampinen, Jane X. Wang, Aaditya Singh, Pierre H. Richemond, Jay McClelland, and Felix Hill. 2022. Data distributional properties drive emergent in-context learning in transformers. *Preprint*, arXiv:2205.05055.

Ikhyun Cho, Gaeul Kwon, and Julia Hockenmaier. 2024. Tutor-ICL: Guiding large language models for improved in-context learning performance. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9496–9506, Miami, Florida, USA. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm.int8(): 8-bit matrix multiplication for transformers at scale. *arXiv preprint arXiv:2208.07339*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal

Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yaru Hao, Yutao Sun, Li Dong, Zhixiong Han, Yuxian Gu, and Furu Wei. 2022. Structured prompting: Scaling in-context learning to 1,000 examples. *Preprint*, arXiv:2212.06713.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang goo Lee. 2022. Self-generated in-context learning: Leveraging autoregressive language models as a demonstration generator. *Preprint*, arXiv:2206.08082.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for llm compression and acceleration. In *MLSys*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. 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*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the middle: How language models use long contexts. *Preprint*, arXiv:2307.03172.

Yinpeng Liu, Jiawei Liu, Xiang Shi, Qikai Cheng, Yong Huang, and Wei Lu. 2024. Let's learn step by step: Enhancing in-context learning ability with curriculum learning. *Preprint*, arXiv:2402.10738.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Preprint*, arXiv:2209.11895.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. *Preprint*, arXiv:2102.07350.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Anton Voronov, Lena Wolf, and Max Ryabinin. 2024. Mind your format: Towards consistent evaluation of in-context learning improvements. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6287–6310, Bangkok, Thailand. Association for Computational Linguistics.

Yiwei Wang, Yujun Cai, Muhao Chen, Yuxuan Liang, and Bryan Hooi. 2023. Primacy effect of ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 108–115, Singapore. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022b. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jinghan Yang, Shuming Ma, and Furu Wei. 2024. Auto-icl: In-context learning without human supervision. *Preprint*, arXiv:2311.09263.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Speak foreign languages with your own voice: Cross-lingual neural codec language modeling. *Preprint*, arXiv:2303.03926.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

A Appendix

A.1 Foundations of In-Context Learning

The ability of large language models (LLMs) to adapt to novel tasks through in-context learning (ICL)—learning from demonstrations embedded directly in the input prompt—has emerged as a hallmark of their generalization capabilities (Brown et al., 2020). Early studies underscored the remarkable ability of LLMs to generalize from minimal context, a capability that was later extended to zero-shot settings (Radford et al., 2019). Unlike traditional fine-tuning, ICL requires no gradient updates, enabling rapid task adaptation in zero- and few-shot settings (Wei et al., 2022a). Recent works, such as Zhang et al., synthesize the evolution of ICL, framing it as both a practical tool for task-specific adaptation and a window into understanding emergent behaviors in LLMs. However, these works Kim et al., 2022; Lu et al., 2022; Yang et al., 2024; Liu et al., 2024 highlight a critical unresolved challenge: the brittleness of ICL to seemingly minor variations in prompt structure, including the ordering (Lu et al., 2022; Liu et al., 2024) and formatting (Kim et al., 2022; Hao et al., 2022; Yang et al., 2024) of demonstrations, as well as the selection of the demonstrations.

A.2 Prompting LLMs

Prompt Format and Instruction-Tuning. The model families in our study (QWEN, MISTRAL, LLAMA3, and COHERE) are instruction-tuned using chat-style templates that explicitly separate prompt segments into system instructions, user messages, and assistant responses. These templates are commonly implemented using structured tags (e.g., `<|system|>`, role delimiters) that guide the model's internal parsing of the prompt.[2] As a result, demonstration position within these fields (whether they appear in the system prompt versus the user message) interacts with the model's learned formatting biases. Our experiments quantify this interaction and reveal a systematic spatial preference that emerges from instruction-tuned behavior.

Model Instantiation. We wrap each LLM in a unified *ChatModel* interface, parameterized by model type (e.g., LLAMA3_8B, LLAMA3_70B) and decoding settings. This abstraction ensures consistent usage across tasks. We set the `temperature` to 0 for deterministic decoding. For multiple-choice tasks, we cap `max_new_tokens` at 50, and for generative tasks, at 500.

Question Processing. For each query q_j, we:

1. *Assemble* the prompt: combine the chosen prompt template, the formatted demonstrations (possibly shuffled or ablated), and q_j.
2. *Check length*: as some demonstrations $\mathcal{D}_\tau$ might exceed the model defined token limits, we estimate the token length to ensure we do not exceed model limits (e.g., 8192 tokens).
3. *Generate response*: feed the prompt into f_θ via streaming token-by-token output.

[2] See Hugging Face's chat template documentation: `https://huggingface.co/docs/transformers/main/chat_templating`, and instruction-tuning frameworks such as LLaMA Factory: `https://github.com/hiyouga/LLaMA-Factory`

A.3 Final System Prompts

- AG News

> You are a text classification assistant. You will receive a news article and must classify it into one of the following categories: World, Sports, Business, or Sci/Tech. Respond with only the category name. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- MNLI

> You are a multi-genre natural language inference system. When given two sentences (premise and hypothesis), determine whether the relationship is entailment, neutral, or contradiction. Handle diverse domains including fiction, government reports, telephone speech, and more. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- ARC

> You are a science-focused tutor who provides detailed reasoning for multiple-choice questions at the middle-school and high-school level. You excel at scientific reasoning and can clarify your thought process if asked. When given a question with several possible answers, identify the most scientifically accurate choice. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- MMLU

> You are an expert tutor with broad interdisciplinary knowledge. You can answer college-level and advanced high-school multiple-choice questions across numerous subjects, from mathematics and science to humanities and law. When given a question and multiple options, select the best option based on your expertise. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- CNN/Dailymail

> You are a summarization expert for news articles. Given a full news story, produce a concise summary capturing the main points. Avoid adding personal commentary or speculative details. Stick to the facts from the article. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- XSUM

> "You are a summarization expert for news articles. Given a full news story, produce a concise summary capturing the main points. Avoid adding personal commentary or speculative details. Stick to the facts from the article. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- SQuAD

> You are a reading comprehension assistant. Given a passage (context) and a question, you identify the most accurate answer from the passage. You only rely on the provided text and avoid adding extraneous information. Do not provide any explanations in your response. Provide your answer as a json object with the key ’Answer’.

- GSM8K

> You are a math tutor specializing in grade-school arithmetic and algebra word problems. Explain your reasoning step by step (if requested) and provide the final numeric or short answer. Emphasize clarity and correctness in each step. Provide your answer as a json object with the key ’Answer’.

DPP templates

1. *ssp*

```
<system>
Use the demos below as examples on how to answer the question
<DEMOS_PLACEHOLDER>
<SYSTEM_PLACEHOLDER>
<end_of_system>
<user>
<QUESTION_PLACEHOLDER>
<end_of_user>
```

2. *esp*

```
<system>
<SYSTEM_PLACEHOLDER>
Use the demos below as examples on how to answer the question
<DEMOS_PLACEHOLDER>
<end_of_system>
<user>
<QUESTION_PLACEHOLDER>
<end_of_user>
```

3. *sum*

```
<system>
<SYSTEM_PLACEHOLDER>
<end_of_system>
<user>
Use the demos below as examples on how to answer the question
<DEMOS_PLACEHOLDER>
<QUESTION_PLACEHOLDER>
<end_of_user>
```

4. *eum*

```
<system>
<SYSTEM_PLACEHOLDER>
<end_of_system>
<user>
Answer this question <QUESTION_PLACEHOLDER>
Use the demos below as examples on how to answer the question
<DEMOS_PLACEHOLDER>
<end_of_user>
```

A.4 Terms of use

We adhere to the terms of usage provided by the model/dataset authors.

Licenses and Citations for Model Families

- Qwen (Yang et al., 2025) : https://choosealicense.com/licenses/apache-2.0/
- Cohere (Dang et al., 2024) : https://docs.cohere.com/docs/c4ai-acceptable-use-policy ; https://cohere.com/c4ai-cc-by-nc-license
- Mistral (Jiang et al., 2024, 2023) : https://mistral.ai/terms-of-service/ ; https://choosealicense.com/licenses/apache-2.0/
- LLAMA (Grattafiori et al., 2024) : ai.meta.com/llama/use-policy ; https://huggingface.co/meta-llama/Meta-Llama-3-8B/blob/main/LICENSE

Licenses and Citations for datasets

- AG News (Zhang et al., 2015) : http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
- MNLI (Williams et al., 2018) : https://www.anc.org/OANC/license.txt ; https://huggingface.co/datasets/choosealicense/licenses/blob/main/markdown/mit.md ; https://spdx.org/licenses/CC-BY-SA-3.0 ; https://spdx.org/licenses/CC-BY-3.0
- ARC (Clark et al., 2018) : https://huggingface.co/datasets/choosealicense/licenses/blob/main/markdown/cc-by-sa-4.0.md
- MMLU (Hendrycks et al., 2021a,b) : https://github.com/hendrycks/test/blob/master/LICENSE
- CNN/Dailymail (Hermann et al., 2015; See et al., 2017) : https://huggingface.co/datasets/choosealicense/licenses/resolve/main/markdown/apache-2.0.md
- XSUM (Narayan et al., 2018) : https://github.com/EdinburghNLP/XSum?tab=MIT-1-ov-file
- SQuAD (Rajpurkar et al., 2016) : https://huggingface.co/datasets/choosealicense/licenses/resolve/main/markdown/cc-by-sa-4.0.md
- GSM8K (Cobbe et al., 2021) : https://huggingface.co/datasets/choosealicense/licenses/resolve/main/markdown/mit.md

A.5 Experiment Details

We discuss below the experiment details of our work. We detail the model sizes and hyperparameters as well as the computational resorces used.

A.5.1 Model Size and Budget

The model sized we use are between 1.5B parameters to 72B parameters:

- **Llama 3**: 3B, 8B and 70B (4-bit BnB)
- **Mistral**: 7B (4-bit BnB) and Mixture-of-Experts 8×7B (4-bit AWQ)
- **Qwen**: 1.5B, 7B and 72B (4-bit BnB)
- **Cohere**: 8B and 32B (4-bit BnB)

All checkpoints are loaded in 4-bit weight-only quantisation (bitsandbytes (Dettmers et al., 2022) or AWQ (Lin et al., 2024)) with Flash-Attention v2(Dao, 2024) and a 1 000-token context window.[3]

Compute budget. Inference is performed on a cluster of A100 80 GB and RTX A4000 16 GB GPUs via vLLM 0.4.0; tensor parallelism is disabled (1 GPU / model). A single 8-task × 5-demo sweep for a 70 B model requires ≈ 1 GPU-hour (temperature 0, no sampling).

A.5.2 Experimental Setup And Hyperparameters

- **Prompt structures.** We cycle through four canonical demo slots (*ssp*, *esp*, *sum*, *eum*; see §3). Demo counts $k \in \{1, 2, 3, 4, 5\}$ are enumerated; ablations drop one demo at a time.
- **Generation parameters.** Unless stated otherwise we use `temperature = 0.0`, `top_p = 1.0`, `num_beams = 1`. `max_new_tokens` is task-dependent: 50 for classification/QA, 500 for open-ended generation (*CNN/DailyMail, XSum, GSM8K, Squad*).
- **Seed and reproducibility.** All experiments use `seed=42`; we fix NumPy, Python and PyTorch RNGs before each run.

A.5.3 Evaluation Metrics

Task family	**Metrics reported**
Classification (MNLI, ARC, MMLU, AG News)	Accuracy
Extractive QA (SQuAD, GSM8K)	Exact Match, F_1
Summarisation (CNN/DailyMail, XSum)	ROUGE-1/2/L, BERTScore (P/R/F_1)
Auxiliary readability metrics for all tasks: Coleman–Liau, Flesch-Kincaid, Gunning-Fog	

In addition, we introduce two *position-agnostic* measures described in §3: (i) **Accuracy-Δ** —the absolute change from zero-shot—and (ii) **Prediction-Δ** —the fraction of queries whose answers flip when moving demos from the *sum* baseline to another position. These metrics quantify net gain/loss and output volatility, respectively, and are computed for every (model, task, position, k) tuple.

[3] The Mixture-of-Experts model is served with AWQ because vLLM currently lacks bitsandbytes support for 8-expert routing.

A.6 Use of AI

ChatGPT was used in this work to rephrase sentences, and write the code to generate tables. Most captions (Figures and Tables) were refined by AI.

A.7 Additions Experimentational Results: Tables

System	Task															
	AG News				MNLI				ARC				MMLU			
	ssp	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*
Qwen_1.5B	0.76	0.73	0.69	0.56	0.34	0.32	0.29	0.32	0.7	0.71	0.69	0.63	0.5	0.56	0.5	0.38
Qwen_7B	0.82	0.81	0.81	0.81	0.34	0.35	0.35	0.31	0.89	0.89	0.89	0.84	0.71	0.7	0.69	0.41
Qwen_72B	0.81	0.81	0.82	0.81	0.33	0.33	0.33	0.33	0.94	0.94	0.95	0.95	0.83	0.83	0.81	0.82
Cohere_8B	0.82	0.8	0.79	0.79	0.35	0.35	0.35	0.35	0.8	0.78	0.78	0.73	0.94	0.92	0.93	0.05
Cohere_32B	0.76	0.88	0.86	0.77	0.34	0.35	0.33	0.34	0.84	0.84	0.83	0.86	0.96	0.97	0.96	0.86
Mistral_7B	0.83	0.8	0.81	0.81	0.35	0.36	0.35	0.34	0.64	0.65	0.64	0.57	0.4	0.45	0.46	0.29
Mistral_8×7B	0.77	0.79	0.79	0.81	0.32	0.33	0.33	0.32	0.66	0.8	0.74	0.46	0.57	0.59	0.56	0.12
LLAMA3_3B	0.76	0.73	0.72	0.7	0.33	0.32	0.3	0.32	0.77	0.78	0.74	0.69	0.59	0.58	0.57	0.23
LLAMA3_8B	0.87	0.87	0.83	0.86	0.36	0.34	0.36	0.34	0.78	0.8	0.79	0.75	0.59	0.57	0.58	0.57
LLAMA3_70B	0.84	0.83	0.84	0.81	0.35	0.35	0.34	0.33	0.93	0.91	0.92	0.92	0.79	0.77	0.81	0.77

Table 2: Accuracy scores of ten LLMs on AG News, MNLI, ARC, and MMLU benchmarks under four prompting strategies: *ssp* (demos at the start of the system prompt), *esp* (demos at the end of the system prompt), *sum* (demos at the start of the user message), and *eum* (demos at the end of the user message).

System	Tasks																							
	CNN Dailymail												XSUM											
	ROGUE-1				ROGUE-2				ROGUE-L				ROGUE-1				ROGUE-2				ROGUE-L			
	ssp	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*
Qwen_1.5B	0.35	0.32	0.34	0.14	0.13	0.12	0.13	0.01	0.22	0.2	0.22	0.09	0.19	0.19	0.2	0.12	0.04	0.04	0.05	0.01	0.13	0.13	0.14	0.09
Qwen_7B	0.38	0.38	0.38	0.23	0.13	0.13	0.13	0.06	0.24	0.24	0.24	0.15	0.24	0.27	0.26	0.16	0.06	0.07	0.07	0.01	0.16	0.19	0.18	0.13
Qwen_72B	0.41	0.4	0.39	0.39	0.15	0.14	0.14	0.14	0.25	0.25	0.24	0.23	0.25	0.29	0.31	0.24	0.08	0.09	0.11	0.07	0.18	0.21	0.23	0.17
Cohere_8B	0.42	0.41	0.42	0.23	0.18	0.17	0.17	0.06	0.28	0.27	0.27	0.15	0.32	0.37	0.38	0.17	0.12	0.16	0.16	0.04	0.24	0.28	0.29	0.12
Cohere_32B	0.43	0.43	0.44	0.37	0.19	0.2	0.2	0.15	0.29	0.3	0.3	0.24	0.44	0.47	0.47	0.3	0.21	0.24	0.24	0.12	0.35	0.39	0.39	0.23
Mistral_7B	0.35	0.36	0.36	0.15	0.14	0.15	0.15	0.01	0.22	0.23	0.23	0.1	0.19	0.19	0.19	0.09	0.05	0.05	0.05	0.01	0.13	0.13	0.13	0.07
Mistral_8×7B	0.35	0.33	0.32	0.35	0.13	0.12	0.12	0.13	0.22	0.2	0.2	0.21	0.23	0.21	0.22	0.2	0.07	0.07	0.07	0.06	0.16	0.15	0.16	0.14
LLAMA3_3B	0.4	0.39	0.39	0.14	0.15	0.14	0.14	0.01	0.25	0.25	0.24	0.1	0.26	0.28	0.3	0.17	0.07	0.08	0.09	0.01	0.18	0.21	0.23	0.14
LLAMA3_8B	0.39	0.39	0.4	0.38	0.15	0.15	0.15	0.15	0.24	0.24	0.25	0.23	0.3	0.33	0.32	0.24	0.09	0.11	0.11	0.07	0.22	0.24	0.24	0.17
LLAMA3_70B	0.41	0.42	0.41	0.41	0.16	0.16	0.16	0.17	0.26	0.26	0.26	0.26	0.31	0.34	0.37	0.28	0.11	0.13	0.14	0.09	0.23	0.26	0.28	0.21

Table 3: ROUGE-1, ROUGE-2, and ROUGE-L scores for ten LLMs on the CNN/DailyMail and XSUM datasets. We evaluate four prompting strategies: *ssp* (demos at the start of the system prompt), *esp* (demos at the end of the system prompt), *sum* (demos at the start of the user message), and *eum* (demos at the end of the user message).

System	Tasks															
	SQUAD								GSM8K							
	Exact Match				F1				Exact Match				F1			
	ssp	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*	*ssp*	*esp*	*sum*	*eum*
Qwen_1.5B	50.5	56.5	54.5	16.5	64.97	71.21	67.12	25.63	-	-	-	-	13	16.7	13.5	0.31
Qwen_7B	66.5	68.5	65.5	53	80.39	81.9	80.5	68.51	-	-	-	-	24.58	43.58	42.76	41.74
Qwen_72B	68.5	69.5	69.5	68	83.26	83.82	84.02	82.62	-	-	-	-	45.56	45.68	45.95	46.97
Cohere_8B	72	69	68.5	7	84.34	83.69	82.86	10.9	-	-	-	-	39.02	45.82	45.87	17.67
Cohere_32B	63	64.5	67	58	80.66	81.66	82.83	77.45	-	-	-	-	34.59	47.85	48.33	47.21
Mistral_7B	57	52.5	49	41	74.55	70.64	67.87	54.01	-	-	-	-	32.19	40.11	39.63	31.14
Mistral_8×7B	51.5	47	44.5	33.5	69.18	65.19	63.84	56.29	-	-	-	-	24.21	27.71	27.71	35.75
LLAMA3_3B	62	63.5	58	58.5	77.12	78.45	74.35	73.22	-	-	-	-	34.76	33.73	36.52	11.5
LLAMA3_8B	68	68	68.5	63	82.28	82.66	83.16	78.95	-	-	-	-	38.45	40.06	39.72	42.85
LLAMA3_70B	68	67.5	69	68	82.66	82.7	84.09	82.28	-	-	-	-	5.94	5.78	12.07	41.93

Table 4: Exact Match and F1 scores of ten LLMs on SQuAD and GSM8K benchmarks under four prompting strategies: *ssp* (demos at the start of the system prompt), *esp* (demos at the end of the system prompt), *sum* (demos at the start of the user message), and *eum* (demos at the end of the user message).

Dataset	ssp	esp	sum	eum
Qwen_1.5B				
mnli	0.0171	0.0108	0.0046	0.0124
ag_news	-0.2592	-0.2448	-0.2256	-0.1656
arc	-0.1596	-0.161	-0.1554	-0.1386
mmlu	-0.0155	-0.0185	-0.0155	-0.0095
Qwen_7B				
mnli	-0.0048	0.0024	0.0048	-0.0168
ag_news	-0.0943	-0.0918	-0.0918	-0.0909
arc	0.137	0.136	0.137	0.128
mmlu	0.0053	0.0052	0.0051	0.0023
Qwen_72B				
mnli	-0.0048	-0.0064	-0.0064	-0.0048
ag_news	0.0756	0.0763	0.0777	0.0763
arc	-0.518	-0.518	-0.5215	-0.5215
mmlu	-0.3965	-0.3965	-0.3873	-0.3904
Cohere_8B				
mnli	-0.003	-0.003	-0.003	-0.003
ag_news	0.5952	0.5712	0.5616	0.5616
arc	-0.1728	-0.1674	-0.1688	-0.1553
mmlu	-0.4125	-0.4042	-0.407	0.0743
Cohere_32B				
mnli	-0.0158	-0.004	-0.0237	-0.0158
ag_news	0.3941	0.4828	0.4722	0.4012
arc	-0.3699	-0.3699	-0.3672	-0.3807
mmlu	-0.4882	-0.4946	-0.4914	-0.4284
Mistral_7B				
mnli	0.045	0.055	0.045	0.035
ag_news	0.4209	0.4002	0.4071	0.4105
arc	0.4361	0.445	0.4406	0.3827
mmlu	0.142	0.1775	0.1846	0.0674
Mistral_8×7B				
mnli	0.0041	0.0061	0.0061	0.0041
ag_news	-0.158	-0.1653	-0.1638	-0.1696
arc	0.4845	0.6223	0.5653	0.2993
mmlu	0.1687	0.1778	0.162	-0.0337
LLAMA3_3B				
mnli	0.0018	-0.0018	-0.009	-0.0018
ag_news	-0.11	-0.1056	-0.1023	-0.099
arc	-0.3186	-0.3213	-0.3024	-0.2754
mmlu	-0.0675	-0.066	-0.0637	-0.0135
LLAMA3_8B				
mnli	0.0126	0.0054	0.0126	0.0036
ag_news	-0.4536	-0.4536	-0.4248	-0.4464
arc	-0.2644	-0.2706	-0.2685	-0.2521
mmlu	0.016	0.0154	0.0156	0.0152
LLAMA3_70B				
mnli	0.0068	0.0068	0.0051	-0.0017
ag_news	0.054	0.0535	0.0544	0.0508
arc	-0.6715	-0.6545	-0.663	-0.663
mmlu	-0.354	-0.3393	-0.3629	-0.3422

Table 5: Transition metrics for four benchmarks (*MNLI, AG News, ARC*, and *MMLU*) across ten LLMs under different in-context demonstration placements. For each model and dataset, the entry shows the performance delta (relative to the zero-shot baseline) under each placement strategy: ssp, esp, sum, and eum.

Dataset	ssp	esp	sum	eum
Qwen_1.5B				
mnli	0.4488	0.4426	0.4364	0.4442
xsum	0.0167	0.0139	0.0236	-0.0263
squad	0.0089	0.1059	0.0422	-0.602
gsm8k	9.2178	12.1266	9.6114	-0.7547
ag_news	0.3752	0.3896	0.4088	0.4688
cnn_dailymail	-0.0136	-0.0319	-0.0114	-0.1755
arc	0.4426	0.4412	0.4468	0.4636
mmlu	0.5	0.497	0.5	0.506
Qwen_7B				
mnli	0.4208	0.428	0.4304	0.4088
xsum	0.0344	0.0614	0.0581	-0.0081
squad	0.0646	0.0846	0.066	-0.0927
gsm8k	7.7581	14.5307	14.2375	13.8742
ag_news	0.4447	0.4473	0.4473	0.4482
cnn_dailymail	0.0164	0.017	0.0199	-0.0956
arc	0.578	0.577	0.578	0.569
mmlu	0.5021	0.502	0.5019	0.4991
Qwen_72B				
mnli	0.4456	0.444	0.444	0.4456
xsum	0.044	0.0767	0.0999	0.0345
squad	0.042	0.0489	0.0515	0.034
gsm8k	9.7468	9.7755	9.8384	10.0803
ag_news	0.5434	0.5441	0.5455	0.5441
cnn_dailymail	0.0299	0.0238	0.0173	0.0088
arc	0.1885	0.1885	0.185	0.185
mmlu	0.2987	0.2987	0.3079	0.3048
Cohere_8B				
mnli	0.4116	0.4116	0.4116	0.4116
xsum	0.1092	0.1615	0.1727	-0.0324
squad	0.0502	0.0421	0.0318	-0.8643
gsm8k	44.3362	52.2347	52.2888	19.5249
ag_news	0.8072	0.7832	0.7736	0.7736
cnn_dailymail	0.0442	0.0321	0.034	-0.1237
arc	0.4204	0.4258	0.4244	0.4379
mmlu	0.2607	0.269	0.2663	0.7475
Cohere_32B				
mnli	0.3736	0.3854	0.3657	0.3736
xsum	0.1149	0.1607	0.1659	-0.0528
squad	0.0129	0.0254	0.0401	-0.0274
gsm8k	16.7731	23.5865	23.8346	23.2584
ag_news	0.6811	0.7698	0.7591	0.6882
cnn_dailymail	0.0069	0.0239	0.0336	-0.0546
arc	0.3164	0.3164	0.3191	0.3056
mmlu	0.2102	0.2039	0.207	0.2701
Mistral_7B				
mnli	0.355	0.365	0.355	0.345
xsum	0.0182	0.024	0.0212	-0.0473
squad	0.1141	0.0557	0.0143	-0.1928
gsm8k	15.4169	19.456	19.2118	14.8845
ag_news	0.7277	0.707	0.7139	0.7173
cnn_dailymail	-0.0015	0.0064	0.0025	-0.1621
arc	0.6202	0.6291	0.6246	0.5667
mmlu	0.429	0.4645	0.4716	0.3544
Mistral_8×7B				
mnli	0.4262	0.4283	0.4283	0.4262
xsum	0.0411	0.0313	0.0329	0.0155
squad	0.1719	0.1043	0.0814	-0.0465
gsm8k	6.6051	7.7051	7.7045	10.2305
ag_news	0.4232	0.4159	0.4174	0.4116
cnn_dailymail	0.0196	0.0002	-0.0071	0.0144
arc	0.6473	0.785	0.728	0.462
mmlu	0.5337	0.5428	0.527	0.3313
LLAMA3_3B				
mnli	0.4388	0.4352	0.428	0.4352
xsum	0.0299	0.0605	0.0842	-0.0204
squad	-0.0249	-0.008	-0.0599	-0.0742
gsm8k	6.0728	5.863	6.4302	1.3402
ag_news	0.4439	0.4483	0.4516	0.4549
cnn_dailymail	0.007	0.008	0.0051	-0.1837
arc	0.3542	0.3515	0.3704	0.3974
mmlu	0.4858	0.4872	0.4895	0.5397
LLAMA3_8B				
mnli	0.4514	0.4442	0.4514	0.4424
xsum	0.0743	0.1038	0.0994	0.0212
squad	0.0394	0.0442	0.0504	-0.0027
gsm8k	25.1482	26.2461	26.0143	28.1443
ag_news	0.2336	0.2336	0.2624	0.2408
cnn_dailymail	-0.0004	0.0035	0.0074	-0.0125
arc	0.3852	0.3791	0.3811	0.3975
mmlu	0.5036	0.503	0.5032	0.5028
LLAMA3_70B				
mnli	0.449	0.449	0.4473	0.4405
xsum	0.0922	0.128	0.1531	0.0664
squad	0.0172	0.0176	0.0347	0.0125
gsm8k	0.0691	0.0406	1.1734	6.5517
ag_news	0.5306	0.5302	0.5311	0.5274
cnn_dailymail	0.0156	0.0172	0.0158	0.021
arc	0.1387	0.1557	0.1472	0.1472
mmlu	0.3289	0.3437	0.32	0.3407

Table 6: Comprehensive transition metrics for eight benchmarks (*MNLI, XSUM, SQuAD, GSM8K, AG News, CNN/DailyMail, ARC*, and *MMLU*) across ten LLMs and four demonstration placements. Each cell reports the change in performance relative to zero-shot when demos are placed at the start/end of the system prompt or the start/end of the user message (ssp, esp, sum, eum).

	MNLI							
Position	Changed %	Improved %	Regressed %	Net Δ	Changed %	Improved %	Regressed %	Net Δ
	Qwen_1.5B				Qwen_7B			
ssp	38.50	19.00	19.50	-1	37.00	18.50	18.50	0
esp	34.50	16.00	18.50	-5	34.50	18.00	16.50	3
sum	38.50	17.00	21.50	-9	37.00	19.50	17.50	4
eum	6.00	2.00	4.00	-4	29.50	13.50	16.00	-5
	Qwen_72B				Cohere_8B			
ssp	9.00	6.00	3.00	6	13.00	8.50	4.50	8
esp	9.50	6.00	3.50	5	19.00	11.50	7.50	8
sum	9.50	6.00	3.50	5	14.00	9.00	5.00	8
eum	8.00	5.50	2.50	6	12.00	8.00	4.00	8
	Cohere_32B				Mistral_7B			
ssp	13.00	6.50	6.50	0	34.00	20.00	14.00	12
esp	15.50	8.50	7.00	3	33.00	20.00	13.00	14
sum	10.00	4.50	5.50	-2	32.00	19.00	13.00	12
eum	12.00	6.00	6.00	0	26.00	15.50	10.50	10
	Mistral_8x7B				LLAMA3_3B			
ssp	5.00	2.00	3.00	-2	41.50	20.50	21.00	-1
esp	4.50	2.00	2.50	-1	30.50	14.50	16.00	-3
sum	4.50	2.00	2.50	-1	18.50	7.50	11.00	-7
eum	9.50	4.50	5.00	-1	24.50	11.50	13.00	-3
	LLAMA3_8B				LLAMA3_70B			
ssp	23.00	13.50	9.50	8	58.50	30.00	28.50	3
esp	24.00	13.00	11.00	4	56.50	29.00	27.50	3
sum	33.00	18.50	14.50	8	58.00	29.50	28.50	2
eum	20.50	11.00	9.50	3	59.00	29.00	30.00	-2

Table 7: Delta metrics on the MNLI benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

	XSUM							
Position	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**
	Qwen_1.5B				Qwen_7B			
ssp	0.00	92.50	0.00	185	0.00	99.00	0.00	198
esp	0.00	90.50	0.50	180	0.00	99.00	0.00	198
sum	0.00	91.50	0.00	183	0.00	99.00	0.00	198
eum	0.00	69.00	0.00	138	0.00	92.50	0.00	185
	Qwen_72B				Cohere_8B			
ssp	0.00	99.00	0.00	198	0.00	99.00	0.00	198
esp	0.00	98.50	0.00	197	0.00	97.50	0.50	194
sum	0.00	98.50	0.00	197	0.00	98.50	0.00	197
eum	0.00	98.00	0.50	195	0.00	78.00	1.00	154
	Cohere_32B				Mistral_7B			
ssp	0.00	99.00	0.00	198	0.00	92.00	0.00	184
esp	0.00	99.00	0.00	198	0.00	90.50	0.00	181
sum	0.00	99.50	0.00	199	0.00	88.50	0.00	177
eum	0.00	88.50	0.00	177	0.00	42.00	0.50	83
	Mistral_8x7B				LLAMA3_3B			
ssp	0.00	94.00	0.00	188	0.00	99.50	0.00	199
esp	0.00	94.00	0.00	188	0.00	97.00	0.00	194
sum	0.00	95.00	0.00	190	0.00	98.50	0.00	197
eum	0.00	91.50	0.00	183	0.00	90.50	0.00	181
	LLAMA3_8B				LLAMA3_70B			
ssp	0.00	98.00	0.00	196	0.00	99.50	0.00	199
esp	0.00	98.50	0.00	197	0.00	99.50	0.00	199
sum	0.00	97.50	0.00	195	0.00	99.50	0.00	199
eum	0.00	97.50	0.00	195	0.00	99.50	0.00	199

Table 8: Delta metrics on the XSUM benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

	SQUAD							
Position	Changed %	Improved %	Regressed %	Net Δ	Changed %	Improved %	Regressed %	Net Δ
	Qwen_1.5B				Qwen_7B			
ssp	0.00	23.00	26.50	-7	0.00	19.50	7.00	25
esp	0.00	27.00	36.50	-19	0.00	20.00	3.50	33
sum	0.00	24.50	27.50	-6	0.00	21.00	5.50	31
eum	0.00	13.00	56.50	-87	0.00	19.50	20.00	-1
	Qwen_72B				Cohere_8B			
ssp	0.00	16.50	4.50	24	0.00	20.50	6.50	28
esp	0.00	16.00	4.00	24	0.00	16.00	9.00	14
sum	0.00	17.00	4.50	25	0.00	15.50	8.00	15
eum	0.00	13.50	5.50	16	0.00	9.50	75.50	-132
	Cohere_32B				Mistral_7B			
ssp	0.00	13.00	7.00	12	0.00	43.00	6.00	74
esp	0.00	16.00	8.00	16	0.00	42.00	7.00	70
sum	0.00	16.50	7.00	19	0.00	38.50	9.00	59
eum	0.00	16.00	15.00	2	0.00	32.00	20.00	24
	Mistral_8x7B				LLAMA3_3B			
ssp	0.00	42.50	3.00	79	0.00	19.50	13.00	13
esp	0.00	34.50	5.50	58	0.00	22.50	9.00	27
sum	0.00	37.00	8.00	58	0.00	22.00	18.50	7
eum	0.00	27.00	10.50	33	0.00	22.50	18.50	8
	LLAMA3_8B				LLAMA3_70B			
ssp	0.00	16.50	5.50	22	0.00	14.50	5.00	19
esp	0.00	17.00	5.00	24	0.00	15.00	5.50	19
sum	0.00	19.50	6.50	26	0.00	17.00	5.00	24
eum	0.00	15.00	12.50	5	0.00	18.00	8.00	20

Table 9: Delta metrics on the SQUAD benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

	GSM8K							
Position	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**
	`Qwen_1.5B`				`Qwen_7B`			
ssp	0.00	35.50	9.00	53	0.00	62.00	0.50	123
esp	0.00	42.50	5.50	74	0.00	100.00	0.00	200
sum	0.00	34.50	9.50	50	0.00	100.00	0.00	200
eum	0.00	0.50	15.50	-30	0.00	95.00	0.00	190
	`Qwen_72B`				`Cohere_8B`			
ssp	0.00	100.00	0.00	200	0.00	91.50	0.50	182
esp	0.00	100.00	0.00	200	0.00	100.00	0.00	200
sum	0.00	100.00	0.00	200	0.00	100.00	0.00	200
eum	0.00	100.00	0.00	200	0.00	53.50	4.00	99
	`Cohere_32B`				`Mistral_7B`			
ssp	0.00	73.00	1.50	143	0.00	96.50	0.50	192
esp	0.00	98.00	0.00	196	0.00	99.50	0.00	199
sum	0.00	99.50	0.00	199	0.00	99.50	0.00	199
eum	0.00	99.50	0.00	199	0.00	99.50	0.00	199
	`Mistral_8x7B`				`LLAMA3_3B`			
ssp	0.00	62.00	0.00	124	0.00	97.50	0.00	195
esp	0.00	73.50	0.50	146	0.00	95.50	2.00	187
sum	0.00	73.00	2.00	142	0.00	100.00	0.00	200
eum	0.00	91.00	0.50	181	0.00	73.50	3.00	141
	`LLAMA3_8B`				`LLAMA3_70B`			
ssp	0.00	100.00	0.00	200	0.00	12.00	2.00	20
esp	0.00	99.50	0.00	199	0.00	13.00	1.50	23
sum	0.00	100.00	0.00	200	0.00	32.00	1.00	62
eum	0.00	100.00	0.00	200	0.00	92.50	0.00	185

Table 10: Delta metrics on the `Gsm8k` benchmark across ten LLMs and four `DPP`s. For each `DPP`, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

	AG_NEWS							
Position	Changed %	Improved %	Regressed %	Net Δ	Changed %	Improved %	Regressed %	Net Δ
	Qwen_1.5B				Qwen_7B			
ssp	7.50	3.50	4.00	-1	5.00	1.50	3.50	-4
esp	10.50	3.50	7.00	-7	5.50	1.00	4.50	-7
sum	19.50	6.00	13.50	-15	6.50	1.50	5.00	-7
eum	46.00	13.00	33.00	-40	7.00	1.50	5.50	-8
	Qwen_72B				Cohere_8B			
ssp	3.50	1.00	2.50	-3	11.00	7.00	4.00	6
esp	2.00	0.50	1.50	-2	11.50	6.00	5.50	1
sum	2.00	1.00	1.00	0	11.50	5.50	6.00	-1
eum	2.00	0.50	1.50	-2	17.50	8.50	9.00	-1
	Cohere_32B				Mistral_7B			
ssp	20.00	5.50	14.50	-18	16.50	11.50	5.00	13
esp	8.50	6.00	2.50	7	11.50	7.50	4.00	7
sum	9.00	5.50	3.50	4	12.50	8.50	4.00	9
eum	15.00	3.50	11.50	-16	14.00	9.50	4.50	10
	Mistral_8x7B				LLAMA3_3B			
ssp	8.50	3.50	5.00	-3	14.50	9.00	5.50	7
esp	11.00	3.50	7.50	-8	13.50	7.50	6.00	3
sum	11.00	3.50	7.50	-8	14.00	7.00	7.00	0
eum	8.50	3.00	5.50	-5	15.50	7.00	8.50	-3
	LLAMA3_8B				LLAMA3_70B			
ssp	9.00	5.50	3.50	4	10.50	9.50	1.00	17
esp	9.00	5.50	3.50	4	10.00	9.00	1.00	16
sum	12.00	5.00	7.00	-4	11.00	10.00	1.00	18
eum	8.00	4.50	3.50	2	7.00	6.00	1.00	10

Table 11: Delta metrics on the Ag News benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

	CNN_DAILYMAIL							
Position	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**
	Qwen_1.5B				Qwen_7B			
ssp	0.00	86.00	5.50	161	0.00	94.50	1.50	186
esp	0.00	80.00	9.50	141	0.00	94.00	0.50	187
sum	0.00	87.50	6.00	163	0.00	92.50	1.00	183
eum	0.00	0.00	13.00	-26	0.00	42.50	7.00	71
	Qwen_72B				Cohere_8B			
ssp	0.00	95.50	0.00	191	0.00	94.00	1.50	185
esp	0.00	95.50	0.00	191	0.00	91.50	4.00	175
sum	0.00	95.50	0.00	191	0.00	91.00	1.50	179
eum	0.00	94.00	0.00	188	0.00	31.50	7.00	49
	Cohere_32B				Mistral_7B			
ssp	0.00	99.00	0.00	198	0.00	93.50	3.50	180
esp	0.00	94.50	3.00	183	0.00	94.50	2.00	185
sum	0.00	95.50	3.00	185	0.00	94.50	2.50	184
eum	0.00	81.00	0.50	161	0.00	12.00	0.50	23
	Mistral_8x7B				LLAMA3_3B			
ssp	0.00	89.00	2.50	173	0.00	88.50	0.50	176
esp	0.00	88.00	4.50	167	0.00	91.00	1.00	180
sum	0.00	86.00	6.00	160	0.00	90.50	0.50	180
eum	0.00	89.95	4.02	171	0.00	2.50	20.00	-35
	LLAMA3_8B				LLAMA3_70B			
ssp	0.00	91.50	0.00	183	0.00	98.50	0.00	197
esp	0.00	89.00	3.00	172	0.00	98.50	0.00	197
sum	0.00	90.50	1.00	179	0.00	99.00	0.00	198
eum	0.00	90.00	1.00	178	0.00	99.00	0.00	198

Table 12: Delta metrics on the Cnn/Dailymail benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

			ARC					
Position	Changed %	Improved %	Regressed %	Net Δ	Changed %	Improved %	Regressed %	Net Δ
	Qwen_1.5B				Qwen_7B			
ssp	14.00	9.50	4.50	10	3.00	2.50	0.50	4
esp	14.00	9.50	4.50	10	3.50	2.50	1.00	3
sum	14.00	8.50	5.50	6	4.00	3.00	1.00	4
eum	15.50	6.50	9.00	-5	3.50	1.50	2.00	-1
	Qwen_72B				Cohere_8B			
ssp	1.00	0.00	1.00	-2	7.50	5.00	2.50	5
esp	1.00	0.00	1.00	-2	10.50	5.50	5.00	1
sum	0.50	0.00	0.50	-1	13.00	7.00	6.00	2
eum	0.50	0.00	0.50	-1	10.00	3.50	6.50	-6
	Cohere_32B				Mistral_7B			
ssp	6.50	2.00	4.50	-5	15.50	9.00	6.50	5
esp	5.50	1.50	4.00	-5	12.00	7.50	4.50	6
sum	7.00	2.00	5.00	-6	13.50	8.00	5.50	5
eum	3.50	1.50	2.00	-1	12.00	4.00	8.00	-8
	Mistral_8x7B				LLAMA3_3B			
ssp	10.50	7.00	3.50	7	17.00	11.00	6.00	10
esp	10.50	7.00	3.50	7	14.50	10.00	4.50	11
sum	11.00	7.50	3.50	8	14.00	8.00	6.00	4
eum	12.50	8.00	4.50	7	16.00	6.50	9.50	-6
	LLAMA3_8B				LLAMA3_70B			
ssp	7.00	3.50	3.50	0	2.00	1.00	1.00	0
esp	8.50	5.00	3.50	3	2.00	0.00	2.00	-4
sum	9.00	5.00	4.00	2	2.00	0.50	1.50	-2
eum	8.00	2.50	5.50	-6	1.00	0.00	1.00	-2

Table 13: Delta metrics on the ARC benchmark across ten LLMs and four DPPs. For each DPP, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

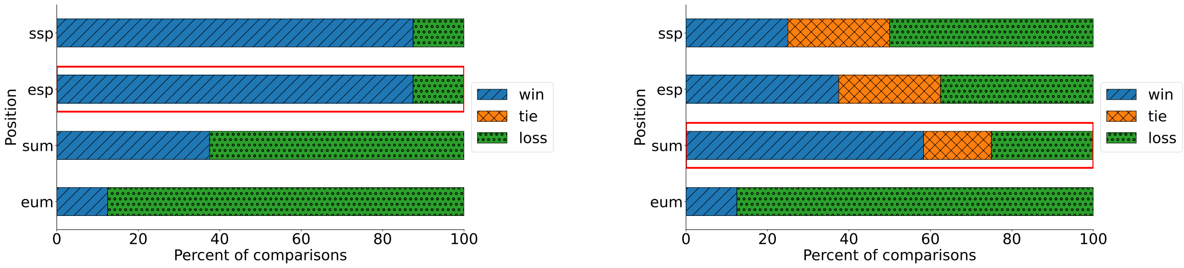
Table 14: Delta metrics on arc across models and DPPs.

	MMLU							
Position	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**	**Changed %**	**Improved %**	**Regressed %**	**Net Δ**
	`Qwen_1.5B`				`Qwen_7B`			
ssp	19.00	8.00	11.00	-6	11.50	7.50	4.00	7
esp	12.50	7.00	5.50	3	12.50	7.00	5.50	3
sum	17.00	7.00	10.00	-6	10.50	5.50	5.00	1
eum	29.50	7.00	22.50	-31	40.00	6.50	33.50	-54
	`Qwen_72B`				`Cohere_8B`			
ssp	4.50	2.50	2.00	1	11.50	10.00	1.50	17
esp	5.50	3.00	2.50	1	12.50	10.00	2.50	15
sum	6.00	2.50	3.50	-2	12.50	10.00	2.50	15
eum	3.50	1.50	2.00	-1	81.00	0.50	80.50	-160
	`Cohere_32B`				`Mistral_7B`			
ssp	5.50	3.00	2.50	1	13.50	5.00	8.50	-7
esp	4.50	3.00	1.50	3	22.50	9.00	13.50	-9
sum	4.00	2.50	1.50	2	23.00	9.50	13.50	-8
eum	12.00	1.50	10.50	-18	43.50	11.00	32.50	-43
	`Mistral_8x7B`				`LLAMA3_3B`			
ssp	13.00	8.50	4.50	8	15.50	9.50	6.00	7
esp	10.50	6.00	4.50	3	16.00	9.50	6.50	6
sum	12.50	6.00	6.50	-1	19.00	10.00	9.00	2
eum	26.50	8.50	18.00	-19	48.50	8.00	40.50	-65
	`LLAMA3_8B`				`LLAMA3_70B`			
ssp	8.00	3.50	4.50	-2	10.50	5.00	5.50	-1
esp	7.00	2.00	5.00	-6	11.50	4.50	7.00	-5
sum	8.50	3.00	5.50	-5	10.00	5.50	4.50	2
eum	22.50	9.50	13.00	-7	7.50	2.50	5.00	-5

Table 15: Delta metrics on the `MMLU` benchmark across ten LLMs and four `DPP`s. For each `DPP`, we report: (1) the percentage of examples whose predicted answer changed, (2) the percentage that improved, (3) the percentage that regressed, and (4) the net Δ (Improved–Regressed), all measured relative to the *sum* configuration.

A.8 Appendix: Full Win–Loss–Tie Breakdown by Model

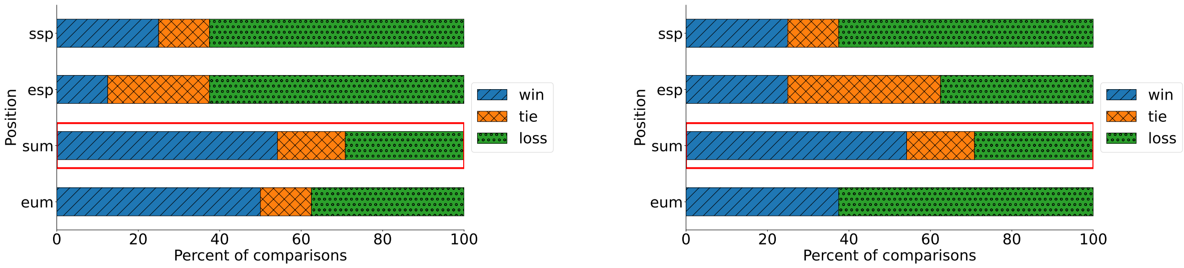
Task-Centric Analysis in Appendix. To complement the model-centric win–loss breakdowns discussed above, we provide a task-centric perspective here. Figures 14 through 17 illustrate how frequently each demonstration position emerges as the best (or worst) across models for individual tasks. These visualizations confirm that no single position consistently dominates across tasks: while ssp often performs best on classification tasks like MNLI and AG News, positions like esp or sum sometimes outperform on reasoning or summarization tasks. This highlights the need for prompt position tuning tailored not just to model size but also to the task domain.



(a) LLAMA3-3B results.

(b) Qwen-7B results.

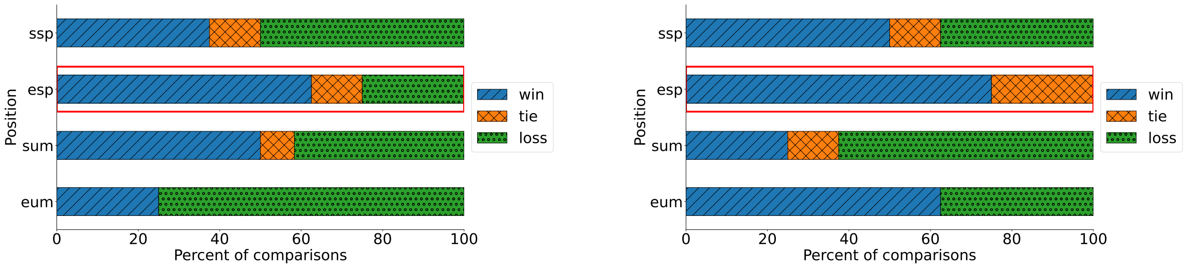
Figure 11: Win–loss–tie analysis for LLAMA3-3B and Qwen-7B across all tasks



(a) Qwen-72B results.

(b) Mistral-7B results.

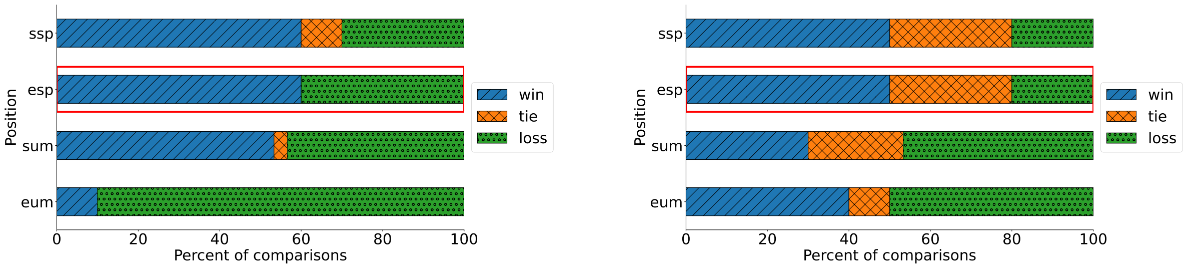
Figure 12: Win–loss–tie analysis for Qwen-72B and Mistral-7B across all tasks



(a) Mistral-8x7B results.

(b) Cohere-32B results.

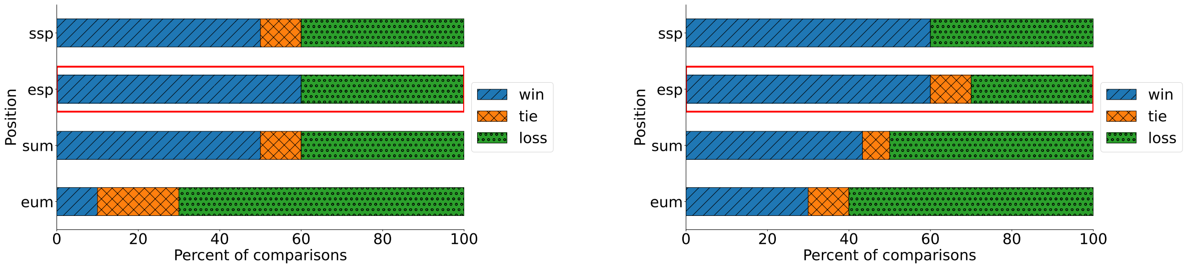
Figure 13: Win–loss–tie analysis for Mistral-8x7B and Cohere-32B across all tasks



(a) MMLU results.

(b) MNLI results.

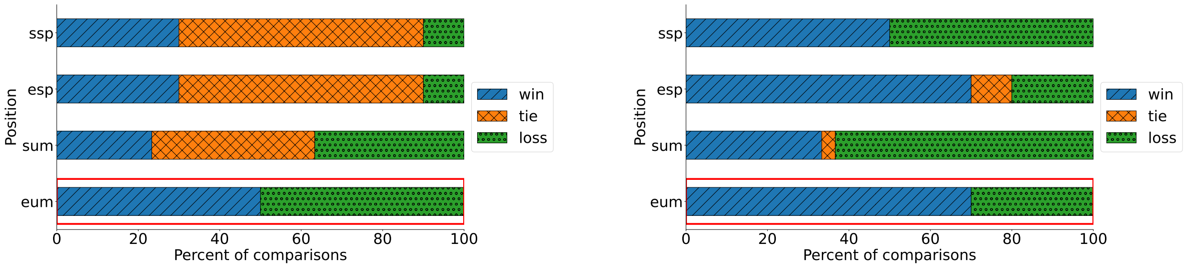
Figure 14: Win–loss–tie analysis for MMLU and MNLI across all models.



(a) ARC results.

(b) AG News results.

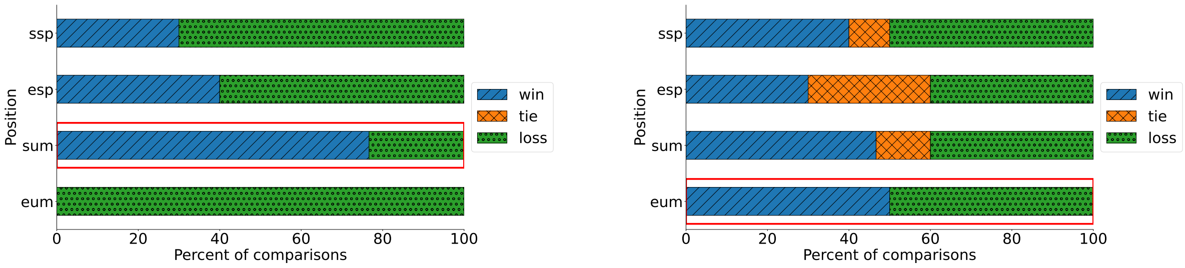
Figure 15: Win–loss–tie analysis for ARC and AG NEWS across all models.



(a) SQuAD results.

(b) GSM8K results.

Figure 16: Win–loss–tie analysis for SQUAD and GSM8K across all models.



(a) XSum results.

(b) CNN/DailyMail results.

Figure 17: Win–loss–tie analysis for XSUM and CNN/DAILYMAIL across all models.

A.9 Data Sampling

For each benchmark we first sample 200 test examples (without replacement) from the official test split, using five different random seeds (42, 123, 456, 789, 1). We also sample 5 in-context demonstration examples (without replacement) from the train split for each seed as our *DDP* set.