BEYOND CONTEXT SELECTION: CONSENSUS OVER RESPONSES TO ALTERNATIVE CONTEXTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models can produce different predictions depending on how inference-time context is composed or applied. Existing approaches have largely addressed this variation by deciding which context to provide before prediction. We instead ask whether the target model's responses to alternative contexts can be used to adapt the final decision to each query. We propose CHORUS, a test-time consensus framework that complements context selection by jointly using responses from alternative ways of composing or applying the supplied context, without training a separate selector. CHORUS combines answers that are repeatedly supported across alternative contexts with answers that receive strong support from a single context. The same principle applies to different demonstration compositions in in-context learning and to different reasoning plans applied to fixed skill documents. Across seven LLMs, 12 classification tasks, six structured and short-form generation tasks, and four skill-augmented reasoning domains, CHORUS shows strong performance against the evaluated selection and consistency baselines. Further analyses show that their relative strengths change with the amount of available context and across tasks, and that adapting consensus decisions to the current query improves prediction. These results support using responses to alternative contexts as test-time evidence beyond context selection alone.

1 INTRODUCTION

Large language models (LLMs) often rely on information and guidance supplied at inference time. Retrieved demonstrations provide examples of how to solve related problems (Rubin et al., 2022), while skill documents supply reusable procedures for specialized tasks (Su et al., 2026). Reasoning plans can further organize how this guidance is applied (Wang et al., 2023a). Providing relevant context, however, does not uniquely determine how the model will use it. The same demonstration pool can be arranged into different sequences, and the same skill documents can be applied through different plans. These alternatives can lead to different responses, making both context selection and context utilization important to the final prediction.

A major line of research improves this process by optimizing the context provided to the model. In in-context learning (ICL), learned retrievers and selection policies use LLM feedback to identify useful demonstrations or demonstration sets (Rubin et al., 2022; Ye et al., 2023; Liu et al., 2024). Such methods address which context to construct, but responses obtained under alternative contexts can also provide evidence for the answer. Existing work explores this opportunity through ensembles over demonstration subsets (Khalifa et al., 2023) and consistency across reasoning paths sampled from a fixed prompt (Wang et al., 2023b; Aggarwal et al., 2023). We build on this perspective by considering responses to alternative ways of composing or applying the supplied context. We ask whether agreement across alternative responses and strong support from an individual response can provide complementary evidence for the answer. We also examine how these signals change as more context becomes available. This complements context selection rather than treating response consensus as an interchangeable context selector.

We propose CHORUS, a test-time consensus framework that uses the target model's responses to the current query without training a separate selector. In ICL, it evaluates different orderings and prefixes of a retrieved demonstration pool. In skill-augmented reasoning, it generates responses under different reasoning plans while keeping the supplied skill documents fixed. We organize these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

responses into sequences as context is progressively added, and represent each sequence in terms of its support for candidate answers. Agreement (AG) favors answers supported across multiple sequences, whereas Commitment (CM) preserves strong support from an individual sequence even when that support is not shared by the others. CHORUS favors sequences whose predictions remain stable and selects a stable point along each sequence before combining the two consensus signals. This design uses both corroboration across alternative contexts and informative individual responses, while allowing the final decision to depend on the current query's response patterns.

We evaluate CHORUS across seven LLMs on ICL classification, structured and short-form generation, and skill-augmented reasoning. The evaluation compares CHORUS with demonstration-selection methods in ICL and consistency-based methods in skill-augmented reasoning, while the controlled skill setting fixes the supplied documents to isolate how they are used. Across these settings, CHORUS shows strong performance, and our analyses support the value of combining complementary response signals and adapting consensus decisions to the current query. Together, these results motivate using responses to alternative contexts as test-time evidence beyond context selection alone.

Our contributions are as follows.

- We study response-level consensus as a complement to context selection, using alternative ways of composing demonstrations and applying supplied skill documents.
- We introduce CHORUS, which combines Agreement and Commitment with stability-based selection to integrate shared and individual answer support without training a separate selector.
- Experiments across seven LLMs show strong performance in three evaluation settings, with analyses supporting complementary consensus rules and consensus decisions adapted to the current query.

2 RELATED WORK

Context selection and composition for in-context learning. Prior work on demonstration selection has focused on constructing useful contexts through semantic relevance (Liu et al., 2022), diversity (Levy et al., 2023), and LLM-supervised retrieval. EPR (Rubin et al., 2022) learns from scores of individual demonstrations, while CEIL (Ye et al., 2023) learns from scored demonstration subsets. Later methods extend this idea to sequential selection, ordering, and a variable number of demonstrations, including Se2 (Liu et al., 2024), SeDPO (Ji et al., 2025), BESC (Gao et al., 2025), Adaptive-k (Taguchi et al., 2025), and DR-IKE (Nafee et al., 2025). D.Va (Zhang et al., 2025a) uses the LLM itself to score candidate demonstrations during selection, while EASE (Wu et al., 2024) restricts the search space to make ordered-context optimization practical. These methods primarily use model feedback to decide which context to construct. CHORUS instead uses the target model's responses to alternative compositions of the retrieved evidence at test time, emphasizing not only context selection but also how the selected evidence is composed and utilized.

Skill retrieval and utilization. Reusable skills provide procedural guidance that agents can retrieve and apply to new tasks. Voyager (Wang et al., 2024) maintains and retrieves from an executable skill library, while SkillRet (Kang et al., 2026) and SkillReason (Jiang et al., 2026) study learned skill retrieval from user requests and capability-oriented supervision. SRA-Bench (Su et al., 2026) further separates skill retrieval, incorporation, and application, showing that successful retrieval alone does not guarantee effective use. Reasoning plans provide one mechanism for organizing how guidance is applied, as in Plan-and-Solve (Wang et al., 2023a). Our controlled skill setting complements retrieval-focused work by fixing the annotated gold skill documents and varying how they are applied, isolating utilization after retrieval.

Test-time consensus. Prediction aggregation across alternative demonstration subsets has been studied with probability-based ensembles (Khalifa et al., 2023). Self-Consistency (Wang et al., 2023b) instead samples multiple reasoning paths from a fixed prompt and aggregates their answers, while Adaptive-Consistency (Aggarwal et al., 2023) and early-stopping Self-Consistency (Li et al., 2024d) adapt the amount of fixed-prompt sampling using answer agreement. CHORUS combines a different source of diversity with test-time consensus: its responses arise from alternative context compositions in ICL or alternative plans for applying the same supplied skill documents. This lets the consensus procedure use variation in how context is utilized, rather than relying only on repeated sampling from a fixed prompt.

3 BACKGROUND AND MOTIVATION

In-context learning (ICL) and skill-augmented reasoning provide task guidance to a language model through its context. We consider both settings with the model parameters held fixed.

3.1 IN-CONTEXT LEARNING AND SKILL-AUGMENTED REASONING

In-context learning. Demonstrations show the model how to perform a task through examples included in the prompt (Brown et al., 2020). For example, a sentiment prompt can show "A delightful film" $\rightarrow$ positive and "A tedious film" $\rightarrow$ negative, followed by the unlabeled review "I would watch it again." The model uses the labeled reviews as demonstrations and predicts a label for the new review. In the retrieval-based ICL setting studied here, a retriever selects candidate examples using similarity or a learned score (Rubin et al., 2022). Each demonstration $e_i = (x_i, y_i)$ pairs an input x_i with its known answer y_i. For a new query q, let the ordered retrieved pool be $\mathcal{C}_q = [e_1, \ldots, e_{n_q}]$. We then choose how many examples to include and in what order. The composition specifies which demonstrations are included, while their order specifies how they are presented. An ordering π of the pool positions and the number of demonstrations k give the demonstration sequence $\mathcal{D}_{\pi,k}(q) = [e_{\pi(1)}, \ldots, e_{\pi(k)}]$. Writing F_θ for the fixed language model, its response r_b to the query preceded by this sequence is

$$r_b = F_\theta(q, \mathcal{D}_{\pi,k}(q)) \,. \tag{1}$$

Here, b indexes the ordering and number of demonstrations, with $k = 0$ denoting a response without demonstrations. The demonstrations supply known answers, while the answer to the new query remains unknown during inference. In our experiments, each query starts from a retrieved candidate pool, and we vary how many demonstrations are used and how they are ordered to obtain alternative context-conditioned responses.

Skill-augmented reasoning. Skill documents provide reusable procedures, formulas, and optional tools (Su et al., 2026). Reasoning plans guide the steps used to solve a problem (Wang et al., 2023a). Here, they are fixed instructions that guide how the model reasons and checks its answer. As a simple example, consider a question asking how far a vehicle travels at 60 km/h in 30 minutes. A skill document could state that distance equals speed multiplied by time and that one hour equals 60 minutes. One plan converts 30 minutes to 0.5 hours before calculating, while another estimates the result first and then verifies the calculation. Let d_q denote the supplied documents and ρ_b a reasoning plan. The model generates a solution using both:

$$r_b = F_\theta(q, d_q, \rho_b). \tag{2}$$

In our utilization setting, d_q contains all documents annotated by the benchmark as relevant to q, called the gold documents. We hold d_q fixed for each query and vary the reasoning plan ρ_b to obtain alternative solutions, including a no-plan response with $\rho_b = \emptyset$. In our experiments, every method receives the same annotated skill documents, so differences arise from how the supplied guidance is applied rather than from retrieval. Detailed models, datasets, retrieval procedures, prompts, and evaluation protocols for both settings are given in Appendix B.

3.2 CONTEXT UTILITY VARIES ACROSS MODELS AND QUERIES

The utility of a context depends on both the target model and the current query. The same context can affect different models, or different queries for the same model, in different ways. This matters for methods that learn context utility from model feedback because the learned preference reflects not only the candidate contexts but also the model used to score them.

Model-specific preferences may not transfer. Even with a moderate retrieved pool, the number of possible contexts becomes extremely large once both subset choice and ordering are considered, as illustrated in Table 1. Exhaustively obtaining target-model feedback over this space is therefore impractical. LLM-supervised selectors instead evaluate a restricted or sampled set of candidates (Rubin et al., 2022; Ye et al., 2023). Because this supervision is produced by a particular model, we examine whether the resulting context preferences remain useful for another target model.

We perform this diagnostic in both ICL and skill-augmented reasoning. For each query, we use evaluation labels to identify which tested demonstration count or reasoning plan works best on the source model, and then evaluate the same choice on the target model. In ICL, the choices are the numbers of demonstrations evaluated across five models and 12 classification tasks. In skill-augmented reasoning, we compare Qwen3-4B and Qwen3-8B (Yang et al., 2025a) across four domains while keeping the gold documents and candidate plans fixed. This label-based analysis is used only to diagnose cross-model transfer. Appendix E.1 provides the detailed transfer procedure.

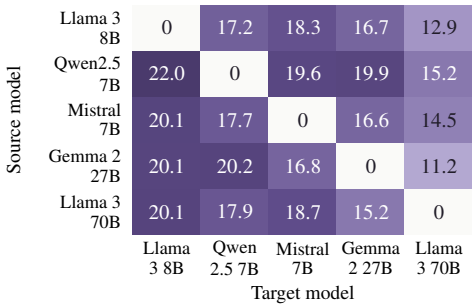
Transferred preferences can substantially reduce performance. Figure 1 shows substantial performance degradation when source-model preferences are applied to a target model in both settings. The skill result is especially informative because both models receive the same documents and candidate plans, so the drop reflects different ways of using the same guidance. Context utility also varies across queries within the same model and task. This means that even a task-level preference for how much context to use may not capture what is useful for an individual query. Appendix E.3 provides query-level examples of this variation.

Motivation. These observations motivate using the target model's responses to the current query directly. Responses to different contexts show how the model's prediction changes across context compositions. Agreement across responses and strong support from an individual response then provide complementary signals for the answer. CHORUS uses these signals for test-time consensus without training a separate utility estimator.

Table 1: Number of possible ICL contexts when selecting k demonstrations from a retrieved pool of n_q candidates. The example uses $n_q = 50$ and $k = 10$.

Context choices	Count	Example
Subsets	$\binom{n_q}{k}$	1.03×10^{10}
Ordered sequences	$n_q!/(n_q-k)!$	3.73×10^{16}

(a) In-context learning



(b) Skill-augmented reasoning

Source model	Target model	Mean accuracy drop
Qwen3-4B	Qwen3-8B	12.57 (87.49%→74.92%)
Qwen3-8B	Qwen3-4B	15.46 (86.76%→71.30%)

Figure 1: Mean accuracy drop under cross-model transfer of (a) numbers of demonstrations and (b) reasoning plans, in percentage points. Parentheses show target-model accuracy using its own best choices → choices transferred from the source model.

4 METHOD

CHORUS uses the target model's responses under alternative ways of composing or applying context as test-time evidence. It first constructs response sequences from these alternatives, represents their support for candidate answers, and then selects and combines this evidence through consensus.

4.1 RESPONSE SEQUENCES AND ANSWER SUPPORT

Sequence construction. CHORUS first constructs multiple response sequences for the current query by varying how the supplied context is composed or applied. Each response sequence is an ordered list of model outputs. We compute answer support at each position, as defined below.

For ICL, let $\mathcal{C}_q = [e_1, \ldots, e_{n_q}]$ denote the retrieved demonstration pool for query q, and let Π be a fixed set of orderings of this pool, whose construction is detailed in Appendix A.1. For an ordering $\pi \in \Pi$ and an evaluated value $k \in \mathcal{K}_q$ for the number of demonstrations, we form

$$\mathcal{D}_{\pi,k}(q) = [e_{\pi(1)}, \ldots, e_{\pi(k)}], \qquad r_{\pi,k} = F_\theta(q, \mathcal{D}_{\pi,k}(q)) \,. \tag{3}$$

The ordered responses

$$\mathbf{r}_\pi^{\mathrm{ICL}}(q) = \left[r_{\pi,k}\right]_{k \in \mathcal{K}_q} \tag{4}$$

define one sequence for each π, where $\mathcal{K}_q$ is ordered by increasing number of demonstrations.

For skill-augmented reasoning, the supplied documents d_q remain fixed. Let $\mathcal{R} = \{\rho_0, \rho_1, \ldots, \rho_P\}$ denote a fixed set of reasoning plans, with $\rho_0 = \emptyset$ representing no additional plan. We independently obtain

$$r_b = F_\theta(q, d_q, \rho_b), \qquad \rho_b \in \mathcal{R}. \tag{5}$$

These responses form a fixed response bank for the query. Let π denote a sequence ordering and $b_{\pi,j}$ the response index at position j. The corresponding sequence is

$$\mathbf{r}_\pi^{\text{skill}}(q) = \left[r_{b_{\pi,1}}, \ldots, r_{b_{\pi,P+1}}\right]. \tag{6}$$

Thus, π changes only how independently generated responses are accumulated, not the responses themselves. In both settings, the resulting sequences provide alternative sequences of evidence for the consensus procedure. Setting-specific sequence construction and orderings are detailed in Appendix A.

Common answer-support representation. The sequences above contain different forms of model output: classification provides scores over predefined labels, whereas generation and skill-augmented reasoning produce free-form responses. We therefore convert every sequence position into a common answer-support representation $p_{\pi,k}(c)$, which measures the support for candidate answer y_c at position k of sequence π.

For ICL classification with labels $\mathcal{Y} = \{y_1, \ldots, y_C\}$, let $\ell_{\pi,k}(c)$ denote the mean conditional token log-probability of candidate label y_c. We define

$$p_{\pi,k}(c) = \frac{\exp \ell_{\pi,k}(c)}{\sum_{c'=1}^{C} \exp \ell_{\pi,k}(c')}. \tag{7}$$

For ICL generation and skill-augmented reasoning, candidate answers are obtained by matching extracted answers across generated responses. Let $a_{\pi,j}$ denote the extracted answer at position j of sequence π, and let $w_{\pi,j}$ denote the confidence weight assigned to that response. In the cumulative representation below, j and k index response positions, rather than numbers of demonstrations. The cumulative support for candidate y_c is

$$p_{\pi,k}(c) = \frac{\sum_{j=1}^{k} w_{\pi,j} \mathbf{1}[a_{\pi,j} \sim y_c]}{\sum_{j=1}^{k} w_{\pi,j} \mathbf{1}[a_{\pi,j} \neq \emptyset]}, \tag{8}$$

where $a_{\pi,j} \sim y_c$ indicates that the extracted answer is grouped with candidate y_c under an answer-equivalence criterion, such as normalized string equality or numeric equivalence. Exact response scoring and equivalence criteria are given in Appendix A.

Before consensus, we normalize each support vector to make candidate support comparable across sequences and sequence positions within the current query. Let $\mu_{\pi,k}$ and $\sigma_{\pi,k}$ denote the mean and standard deviation of $p_{\pi,k}$ across candidate answers:

$$z_{\pi,k}(c) = \frac{p_{\pi,k}(c) - \mu_{\pi,k}}{\sigma_{\pi,k} + \varepsilon}, \qquad \tilde{p}_{\pi,k}(c) = \frac{\exp z_{\pi,k}(c)}{\sum_{c'} \exp z_{\pi,k}(c')}. \tag{9}$$

The normalized support $\tilde{p}_{\pi,k}$ is the common input to the consensus procedure below.

4.2 TEST-TIME CONSENSUS

Given the normalized answer support $\tilde{p}_{\pi,k}$ at each sequence position, CHORUS combines two complementary forms of evidence. Agreement (AG) emphasizes support shared across sequences, while Commitment (CM) preserves strong support from an individual sequence by taking the maximum normalized answer support across sequences.

Agreement (AG). AG first identifies sequences whose top predictions remain stable as evidence accumulates. For each sequence, let $\hat{c}_\pi(k) = \arg\max_c \tilde{p}_{\pi,k}(c)$. We define its sequence-level stability as

$$s_{\text{stab}}(\pi) = \frac{1}{|\mathcal{T}_\pi|} \sum_{(k^-,k) \in \mathcal{T}_\pi} \mathbf{1}\left[\hat{c}_\pi(k) = \hat{c}_\pi(k^-)\right], \tag{10}$$

Table 2: ICL classification performance on Llama 3 8B across 12 tasks. Accuracy is reported in percent. Avg. is the mean accuracy across the 12 tasks.

Method	SST-5	CoLA	MRPC	WiC	CSQA	COPA	CB	WinoG	WSC	OBQA	PIQA	RTE	Avg.
Fixed number of demonstrations													
RANDOM	47.59	50.52	66.91	65.20	74.37	79.00	82.14	58.48	58.65	75.80	74.05	47.65	65.03
TOPK-BM25 (Robertson & Zaragoza, 2009)	51.50	54.75	70.83	67.40	72.15	79.00	83.93	**59.75**	63.46	77.40	76.06	47.29	66.96
TOPK-BERT (Devlin et al., 2019)	51.14	62.01	71.81	64.89	74.77	84.00	83.93	55.88	53.85	78.80	74.86	47.29	66.94
DPP-BERT (Chen et al., 2018)	52.95	57.69	72.55	65.20	**75.10**	82.00	83.93	59.19	55.77	78.20	74.27	47.29	67.01
D.VA (Zhang et al., 2025a)	48.23	55.79	71.08	61.76	71.50	**90.00**	83.93	59.51	57.69	77.60	76.88	49.46	66.95
EPR (Rubin et al., 2022)	**53.77**	55.39	79.17	63.17	72.81	81.00	80.36	56.51	67.31	78.00	75.84	47.29	67.55
TOPICK (Kweon et al., 2025)	47.32	51.27	70.59	63.32	69.78	86.00	82.14	58.33	61.54	**81.00**	**77.69**	50.54	66.63
EASE (Wu et al., 2024)	48.96	55.37	60.05	63.64	72.56	81.00	**87.50**	51.70	63.46	75.60	74.76	47.29	65.16
SE2 (Liu et al., 2024)	43.42	47.51	69.85	60.97	68.71	84.00	78.57	57.70	58.65	76.00	72.69	46.93	63.75
SEDPO (Ji et al., 2025)	48.68	80.35	**79.90**	57.37	70.52	82.00	82.14	58.80	51.92	74.20	74.65	48.74	67.44
Query-adaptive number of demonstrations													
DR-IKE (Nafee et al., 2025)	35.79	41.93	65.93	62.38	69.45	71.00	69.64	56.83	63.46	76.00	68.99	46.57	60.66
ADAPTIVE-k (Taguchi et al., 2025)	37.30	80.20	65.40	62.90	68.10	70.00	69.60	57.10	12.50	74.00	70.90	51.60	59.97
CHORUS	53.13	**85.23**	73.04	**68.03**	74.86	85.00	**87.50**	58.64	**69.23**	78.60	77.04	**52.71**	**71.92**

where $\mathcal{T}_\pi$ contains adjacent evaluated positions of sequence π. We retain the m_{AG} sequences with the largest stability scores, denoted Π_{AG}, and aggregate their support by

$$p_{\mathrm{AG}}(c,k) \propto \left(\prod_{\pi \in \Pi_{\mathrm{AG}}} \tilde{p}_{\pi,k}(c) \right)^{1/|\Pi_{\mathrm{AG}}|}. \tag{11}$$

The geometric mean favors candidates supported consistently across the retained sequences.

Commitment (CM). CM instead preserves strong evidence from any individual sequence. Let Π_{CM} contain all available sequences. At each position,

$$p_{\mathrm{CM}}(c,k) \propto \max_{\pi \in \Pi_{\mathrm{CM}}} \tilde{p}_{\pi,k}(c). \tag{12}$$

Thus, strong support for a candidate can be retained even when it is not shared across sequences.

Stable evidence selection and fusion. Before fusion, we select a stable sequence position for each rule so that the final prediction is based on persistent rather than transient support. Let $k_1 < \cdots < k_M$ denote the evaluated sequence positions, and let $\hat{c}_r(k_i) = \arg\max_c p_r(c, k_i)$ for $r \in \{\mathrm{AG}, \mathrm{CM}\}$. We define the stable positions of rule r as

$$\mathcal{S}_r = \{k_i : \hat{c}_r(k_i) = \hat{c}_r(k_{i-1}) = \cdots = \hat{c}_r(k_{i-W+1})\}, \tag{13}$$

where W is the stability window. AG selects the largest stable position,

$$k^\star_{\mathrm{AG}} = \max \mathcal{S}_{\mathrm{AG}}. \tag{14}$$

For CM, we additionally require a margin between the two highest candidate scores,

$$\Delta_{\mathrm{CM}}(k_i) = p_{\mathrm{CM}}(c_{(1)}, k_i) - p_{\mathrm{CM}}(c_{(2)}, k_i), \tag{15}$$

where $c_{(1)}$ and $c_{(2)}$ denote the highest- and second-highest-scoring candidates. We select

$$k^\star_{\mathrm{CM}} = \max \{k_i \in \mathcal{S}_{\mathrm{CM}} : \Delta_{\mathrm{CM}}(k_i) > \tau\}. \tag{16}$$

If no stable position is found, AG falls back to the position with the strongest top-answer support. For CM, if no position satisfies both stability and margin, we first drop the margin requirement and then use the same fallback if necessary.

We then combine the selected distributions,

$$p_{\mathrm{ens}}(c) = \frac{w_{\mathrm{AG}}\, p_{\mathrm{AG}}(c, k^\star_{\mathrm{AG}}) + w_{\mathrm{CM}}\, p_{\mathrm{CM}}(c, k^\star_{\mathrm{CM}})}{w_{\mathrm{AG}} + w_{\mathrm{CM}}}, \tag{17}$$

and return $\hat{y}(q) = y_{\arg\max_c p_{\mathrm{ens}}(c)}$. As shown in Section 5.3, AG and CM exhibit complementary strengths across tasks, motivating their fixed combination rather than selecting a rule separately for each task. Implementation details are given in Appendix A.

Table 3: Generation performance on Llama 3 8B across six tasks. Scores follow each task's evaluation metric, and Avg. is the unweighted mean across the six tasks.

Method	GeoQuery	MTOP	NL2Bash	Break	SMCalFlow	WebQs	Avg.
Fixed number of demonstrations							
RANDOM	57.5	23.5	19.9	29.5	16.5	3.0	25.0
TOPK-BM25 (Robertson & Zaragoza, 2009)	85.0	72.0	**45.4**	52.5	62.5	8.5	54.3
TOPK-BERT (Devlin et al., 2019)	**88.0**	74.0	38.1	53.0	62.5	12.0	54.6
DPP-BERT (Chen et al., 2018)	83.0	73.0	38.0	53.5	**65.5**	9.0	53.7
D.VA (Zhang et al., 2025a)	69.0	61.5	32.9	38.0	49.5	21.0	45.3
EPR (Rubin et al., 2022)	85.5	**76.5**	43.2	**54.5**	65.0	9.5	55.7
TOPICK (Kweon et al., 2025)	44.0	47.0	38.0	35.0	44.5	12.0	36.8
EASE (Wu et al., 2024)	15.0	4.0	21.3	13.5	9.5	5.5	11.5
SE2 (Liu et al., 2024)	33.0	48.0	39.0	40.5	50.0	16.5	37.8
SEDPO (Ji et al., 2025)	32.0	50.0	35.3	36.0	43.5	22.0	36.5
Query-adaptive number of demonstrations							
DR-IKE (Nafee et al., 2025)	73.0	62.5	35.3	43.5	48.0	10.0	45.4
ADAPTIVE-k (Taguchi et al., 2025)	32.5	51.0	39.9	34.5	41.5	**24.0**	37.2
CHORUS	86.0	73.5	45.2	51.0	63.0	17.0	**56.0**

5 EXPERIMENTS

5.1 IN-CONTEXT LEARNING

We evaluate CHORUS on 12 classification tasks across five LLMs: Llama 3 8B and 70B (Grattafiori et al., 2024), Qwen2.5 7B Instruct (Qwen et al., 2025), Mistral 7B Instruct v0.3 (Jiang et al., 2023), and Gemma 2 27B (Team et al., 2024). We also evaluate six structured and short-form generation tasks with Llama 3 8B. The baselines span retrieval heuristics, learned selectors trained from LLM-scored candidate contexts, and methods that optimize context order or size. We group them by whether they use a fixed number of demonstrations or adapt the number of demonstrations to each query. Prior ICL research has primarily focused on constructing a useful context through demonstration selection, ordering, or adaptive context size, while comparatively less work studies how responses to alternative context compositions can be jointly exploited at test time. The baselines and CHORUS therefore operate at different stages of the ICL pipeline, but their comparison is informative for assessing whether response-level consensus provides additional value beyond context construction alone. Experimental settings and additional results are provided in Appendices B and C.

Classification. Table 2 shows that CHORUS achieves the highest average accuracy on Llama 3 8B, reaching 71.92 compared with 67.55 for the strongest evaluated baseline, with the same trend across four additional LLMs. This comparison highlights that ICL performance is not determined by retrieval alone. Starting from the retrieved candidate demonstrations, different orderings and prefixes can induce different model predictions, and CHORUS uses these response sequences as additional test-time evidence. The gains therefore suggest that how retrieved demonstrations are composed and utilized by the target model is an important part of ICL beyond selecting the demonstrations themselves.

Generation. We next test whether the same consensus formulation extends beyond predefined classification labels to free-form generation. For generated outputs, CHORUS groups equivalent responses into candidate answers and accumulates their support along response sequences before applying the same consensus procedure. Table 3 shows competitive performance across six structured and short-form generation tasks and the highest average score among the evaluated methods. More importantly, these results show that the framework is not tied to classification probabilities: cumulative answer support provides a common representation that allows the same sequence-based consensus to operate on free-form responses.

5.2 SKILL-AUGMENTED REASONING

Comparison under fixed skill information. To isolate skill utilization from retrieval, we give every method the same annotated gold skill documents and evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025a) on 1,546 questions from four SRA-Bench (Su et al., 2026) domains. We compare direct skill injection, fixed-prompt consistency methods that obtain diversity by repeated sampling, and CHORUS,

Table 4: Skill-augmented reasoning performance across response budgets. Results are mean ± SD accuracy over three runs, averaged equally across four domains.

Method	Decoding	Aggregation	Response budget	Qwen3-4B	Qwen3-8B
Best single plan	Greedy	–	1	75.96 ± 0.37	76.39 ± 0.10
Full-Skill Injection (Su et al., 2026)	Greedy	–	1	73.70 ± 0.62	70.64 ± 0.59
(gold context, no plan)	Sampling	–	1	74.33 ± 0.16	72.08 ± 1.14
Self-Consistency	Sampling	Majority	6	77.33 ± 0.34	76.21 ± 0.36
(Wang et al., 2023b)	Sampling	Weighted	6	77.55 ± 0.21	76.43 ± 0.69
Adaptive-Consistency (Aggarwal et al., 2023)	Sampling	Majority	6	77.33 ± 0.34	76.21 ± 0.36
CHORUS	Greedy	AG+CM	6	$\mathbf{78.98 \pm 0.31}$	$\mathbf{79.69 \pm 0.17}$

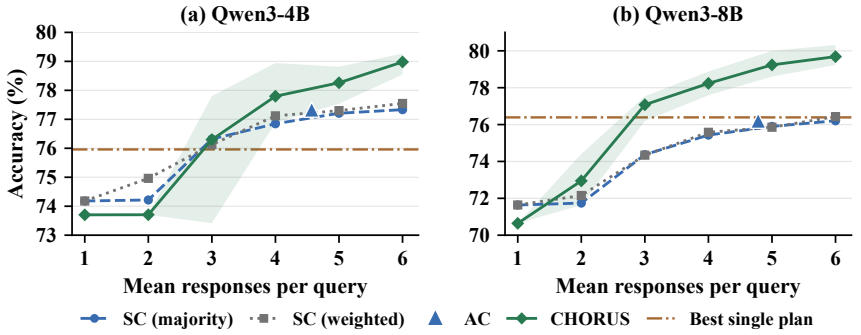


Figure 2: Skill accuracy versus mean responses per query for Qwen3-4B and Qwen3-8B. Shading shows variation across plan-addition orders, and the horizontal line marks Best single plan.

which generates plan-conditioned responses and combines them through test-time consensus. At the matched six-response budget, CHORUS reaches 78.98% on Qwen3-4B and 79.69% on Qwen3-8B, exceeding both Self-Consistency variants and Adaptive-Consistency. Because every method receives the same gold skill documents, this comparison isolates how the supplied information is used rather than what is retrieved. Self-Consistency and Adaptive-Consistency obtain diversity from repeated sampling under a fixed prompt, whereas CHORUS obtains structured diversity by applying different reasoning plans before consensus.

Comparison with the best single plan. Best single plan tests whether the gain can be explained by using one strong reasoning plan. For each model, we retrospectively select the plan with the highest average evaluation accuracy and apply it to every query. It reaches 75.96% on Qwen3-4B and 76.39% on Qwen3-8B, remaining below CHORUS. This comparison indicates that the gain is not captured by selecting a single favorable plan and instead comes from retaining and combining responses produced under multiple plans.

Scaling with additional plan-conditioned responses. Figure 2 examines how performance changes as more plan-conditioned responses become available. CHORUS maintains higher mean accuracy than both Self-Consistency variants at every common fixed budget from four to six responses on both models. At four responses, its mean accuracy is also higher than Adaptive-Consistency, which uses 4.57 and 4.78 responses on average for Qwen3-4B and Qwen3-8B, respectively. The advantage remains as additional plan-conditioned responses are added, suggesting that structured response diversity remains useful as the response sequence grows. Detailed experimental settings are given in Appendix B.3, and domain-level results are reported in Appendix C.2.

5.3 ANALYSIS OF TEST-TIME CONSENSUS

Complementarity across available evidence and tasks. AG and CM capture different forms of evidence. AG emphasizes support that recurs across alternative sequences, favoring answers

Table 5: ICL accuracy with at most k available demonstrations, averaged equally over 12 tasks.

Method	k=3	k=10	k=25	k=50	Full
TopK-BERT (Devlin et al., 2019)	63.45	67.00	68.03	66.94	67.05
TopicK (Kweon et al., 2025)	62.56	66.49	66.48	66.68	67.65
EASE (Wu et al., 2024)	65.79	67.09	65.73	67.48	66.28
AG	68.00	70.60	69.59	70.56	70.64
CM	67.00	69.83	70.82	70.88	71.03
CHORUS	**68.19**	**70.89**	**71.17**	**71.71**	**71.92**

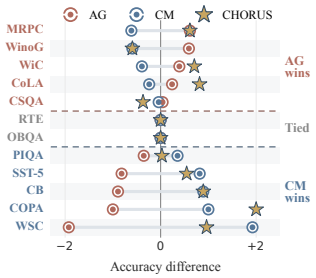


Figure 3: Task-level accuracy of AG, CM, and CHORUS, relative to the average of AG and CM.

corroborated by multiple context compositions. CM instead preserves the strongest support from an individual sequence, allowing one particularly informative composition to remain influential even when the same support is not repeated elsewhere. Table 5 shows that their relative advantage changes as more demonstrations become available within each sequence. AG is stronger at the smaller budgets ($k = 3$ and $k = 10$), whereas CM becomes stronger from $k = 25$ onward. This crossover is consistent with the two rules using evidence differently: with shorter contexts, repeated agreement across sequences can suppress weak or unstable support, while richer contexts can produce a strongly informative individual sequence that CM is designed to preserve. CHORUS combines both signals and achieves the highest accuracy at every reported budget, reaching 71.92 in the full setting.

Figure 3 compares the two rules at the task level by centering each task on the average accuracy of AG and CM. Neither rule is consistently dominant across tasks. However, we observe a descriptive tendency for several tasks with small demonstration pools to favor CM, while most AG-favored tasks have larger pools. The tendency has clear exceptions, indicating that pool size alone does not explain which rule is preferable and further supporting their combination in CHORUS.

Do consensus decisions depend on the current query? We test this by transferring consensus decisions between queries while keeping each query's responses and scores fixed. In ICL, we transfer the sequence subsets and positions used by AG and CM to another query with compatible positions and apply them to that query's own sequences. In skill-augmented reasoning, we analogously transfer the final response index selected by consensus.

Table 6: Accuracy using each query's own consensus decision versus decisions shuffled among queries within the same task.

Setting	Own	Shuffled	Δ
ICL: Llama 3 8B	71.92	70.59	+1.32
Skill: Qwen3-4B	78.98	73.36	+5.62
Skill: Qwen3-8B	79.69	71.94	+7.75

Table 6 shows consistent performance drops when the decision is transferred from another query. Because the underlying responses and scores remain unchanged, the degradation indicates that consensus decisions are meaningfully adapted to the current query rather than reflecting only a task-level preference.

Additional analyses of ordering and subset selection, score normalization and aggregation, and the decision-shuffling protocol are provided in Appendices D and E.2.

6 CONCLUSION

We presented CHORUS, a test-time consensus framework that uses the target model's responses to alternative contexts as additional evidence for the final prediction. Rather than replacing context selection, CHORUS complements it by considering how the model responds when supplied context is composed or applied in different ways. Across classification, structured and short-form generation, and skill-augmented reasoning, CHORUS shows strong performance against the evaluated selection and consistency baselines. Our analyses show that context utility varies across models and queries, while complementary response signals and query-adaptive consensus decisions improve prediction. Together, these results suggest that responses to alternative contexts provide useful test-time evidence beyond context selection alone.

AI USE STATEMENT

Generative AI tools assisted manuscript editing, implementation and debugging of research code, and preparation of scientific figures. All AI-assisted outputs were reviewed and verified by the authors, who take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The appendix provides detailed experimental settings, dataset and evaluation protocols, method implementation details, and additional results and ablations needed to reproduce the reported experiments.

REFERENCES

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 12375–12396. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.761. URL https://aclanthology.org/2023.emnlp-main.761/.

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. Skill-based few-shot selection for in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13472–13492, 2023.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, et al. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571, 2020. URL https://aclanthology.org/2020.tacl-1.36/.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1, 2009.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013. URL https://aclanthology.org/D13-1160/.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

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.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31, 2018.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, 2023. URL https://aclanthology.org/2023.emnlp-main.489/.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pp. 177–190. Springer, 2005.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pp. 107–124, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*, 2005.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. Mitigating label biases for in-context learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14014–14031, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.783. URL https://aclanthology.org/2023.acl-long.783/.

Xiang Gao, Ankita Sinha, and Kamalika Das. Learning to search effective example sequences for in-context learning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6152–6161, 2025. doi: 10.18653/v1/2025.findings-naacl.342. URL https://aclanthology.org/2025.findings-naacl.342/.

Soumya Suvra Ghosal, Soumyabrata Pal, Koyel Mukherjee, and Dinesh Manocha. Promptrefine: Enhancing few-shot performance on low-resource indic languages with example selection from related example banks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 351–365, 2025.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pp. 1–9, 2007.

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.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7, pp. 785–794, 2006.

Yinhan He, Wendy Zheng, Song Wang, Zaiyi Zheng, Yushun Dong, Yaochen Zhu, and Jundong Li. Hierarchical demonstration order optimization for many-shot in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9FAm023EmI.

Brandon Huang, Chancharik Mitra, Leonid Karlinsky, Assaf Arbelle, Trevor Darrell, and Roei Herzig. Multimodal task vectors enable many-shot multimodal in-context learning. *Advances in Neural Information Processing Systems*, 37:22124–22153, 2024.

Yuwen Ji, Luodan Zhang, Ambyerhan, Haoran Que, Lei Shi, Wang Chao, and Yue Zhang. Learning to rank for in-context example retrieval. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=WyQ20adbUb.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Donghong Jiang, Endian Lin, Luoping Cui, Hanqing Liu, Mingjie Liu, Fan Yang, Hong Wang, Zhao Yang, and Chuang Zhu. SkillReason: Reasoning-enhanced agent skill retrieval for implicit user requests. *arXiv preprint arXiv:2608.08640*, 2026. URL https://arxiv.org/abs/2608.08640.

Ryangkyung Kang, Hongcheol Cho, and Youngeun Kim. SkillRet: A large-scale benchmark for skill retrieval in LLM agents. *arXiv preprint arXiv:2605.05726*, 2026. URL https://arxiv.org/abs/2605.05726.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. Exploring demonstration ensembling for in-context learning. *arXiv preprint arXiv:2308.08780*, 2023. URL https://arxiv.org/abs/2308.08780.

Nikhil Khandekar, Qiao Jin, Guangzhi Xiong, Soren Dunn, Serina S Applebaum, Zain Anwar, Maame Sarfo-Gyamfi, Conrad W Safranek, Abid A Anwar, Andrew Zhang, Aidan Gilson, Maxwell B Singer, Amisha Dave, Andrew Taylor, Aidong Zhang, Qingyu Chen, and Zhiyong Lu. MedCalc-Bench: Evaluating large language models for medical calculations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 84730–84745, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/99e81750f3fdfcaf9613db2dbf4bd623-Abstract-Datasets_and_Benchmarks_Track.html.

Jannik Kossen, Yarin Gal, and Tom Rainforth. In-context learning learns label relationships but is not conventional learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=YPIA7bgd5y.

Mayank Kothyari, Sunita Sarawagi, Soumen Chakrabarti, Gaurav Arora, and Srujana Merugu. Diverse in-context example selection after decomposing programs and aligned utterances improves semantic parsing. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5611–5629, 2025.

Wonbin Kweon, SeongKu Kang, Runchu Tian, Pengcheng Jiang, Jiawei Han, and Hwanjo Yu. Topic coverage-based demonstration retrieval for in-context learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 19911–19923, 2025.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. *KR*, 2012(13th):3, 2012.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1401–1422, 2023.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021. URL https://aclanthology.org/2021.eacl-main.257/.

Jiaqian Li, Qisheng Hu, Jing Li, and Wenya Wang. Stare at the structure: Steering icl exemplar selection with structural alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14776–14793, 2025a.

Lvxue Li, Jiaqi Chen, Xinyu Lu, Yaojie Lu, Hongyu Lin, Shuheng Zhou, Huijia Zhu, Weiqiang Wang, Zhongyi Liu, Xianpei Han, and Le Sun. Debiasing in-context learning by instructing LLMs how to follow demonstrations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7203–7215, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.430. URL https://aclanthology.org/2024.findings-acl.430/.

Siqi Li, Danni Liu, and Jan Niehues. Optimizing rare word accuracy in direct speech translation with a retrieval-and-demonstration approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12703–12719, 2024b.

Wei Li, Wen Luo, Guangyue Peng, and Houfeng Wang. Explanation based in-context demonstrations retrieval for multilingual grammatical error correction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4881–4897, 2025b.

Yichuan Li, Xiyao Ma, Sixing Lu, Kyumin Lee, Xiaohu Liu, and Chenlei Guo. MEND: Meta demonstration distillation for efficient and effective in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024c. URL https://openreview.net/forum?id=2Y5kBPtU0o.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *International Conference on Learning Representations*, 2024d. URL https://arxiv.org/abs/2401.10480.

Xi Victoria Lin, Chenglong Wang, Luke Zettlemoyer, and Michael D. Ernst. NL2Bash: A corpus and semantic parser for natural language interface to the linux operating system. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 2018. URL https://aclanthology.org/L18-1491/.

Haoyu Liu, Jianfeng Liu, Shaohan Huang, Yuefeng Zhan, Hao Sun, Weiwei Deng, Furu Wei, and Qi Zhang. se2: Sequential example selection for in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5262–5284, 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.

Yujun Mao, Yoon Kim, and Yilun Zhou. CHAMP: A competition-level dataset for fine-grained analyses of LLMs' mathematical reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13256–13274, 2024. URL https://aclanthology.org/2024.findings-acl.785/.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Mahmud Wasif Nafee, Maiqi Jiang, Haipeng Chen, and Yanfu Zhang. Dynamic retriever for in-context knowledge editing via policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16744–16757, 2025. doi: 10.18653/v1/2025.emnlp-main.848. URL https://aclanthology.org/2025.emnlp-main.848/.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13679–13707, 2024. URL https://aclanthology.org/2024.acl-long.739/.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9090–9101, 2024.

Mohammad Taher Pilehvar and Jose Camacho-Collados. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273, 2019.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of machine learning and systems*, 5:606–624, 2023.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389, 01 2009. doi: 10.1561/1500000019.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pp. 90–95, 2011.

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, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.191. URL https://aclanthology.org/2022.naacl-main.191.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020.

Alexander Scarlatos and Andrew Lan. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*, 2023.

Suzanna Sia, David Mueller, and Kevin Duh. Where does in-context learning happen in large language models? *Advances in Neural Information Processing Systems*, 37:32761–32786, 2024.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Weihang Su, Jianming Long, Qingyao Ai, Qiaozhi He, Yichen Tang, Changyue Wang, Yiteng Tu, Yingbo Wang, and Yiqun Liu. Skill retrieval augmentation for agentic AI. *arXiv preprint arXiv:2604.24594*, 2026. URL https://arxiv.org/abs/2604.24594.

Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. Efficient context selection for long-context qa: No tuning, no iteration, just adaptive-k. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20105–20130, 2025. doi: 10.18653/v1/2025.emnlp-main.1017. URL https://aclanthology.org/2025.emnlp-main.1017/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Chenming Tang, Zhixiang Wang, and Yunfang Wu. Scoi: Syntax-augmented coverage-based in-context example selection for machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9956–9971, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://arxiv.org/abs/2305.16291.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634. Association for Computational Linguistics, 2023a. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

Xubin Wang, Jianfei Wu, Yuan Yichen, Deyu Cai, Mingzhe Li, and Weijia Jia. Demonstration selection for in-context learning via reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=sugs65XoGg.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023b. URL https://arxiv.org/abs/2203.11171.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198, 2020. URL https://aclanthology.org/2020.tacl-1.13/.

Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with ease? efficient ordering-aware automated selection of exemplars. *Advances in Neural Information Processing Systems*, 37:122706–122740, 2024.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. URL https://arxiv.org/abs/2505.09388.

Haolin Yang, Hakaze Cho, Yiqiao Zhong, and Naoya Inoue. Unifying attention heads and task vectors via hidden state geometry in in-context learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=FIfjDqjV0B.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pp. 39818–39833. PMLR, 2023.

John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1050–1055, 1996. URL https://www.cs.utexas.edu/~ai-lab/pub-view.php?PubID=51437.

Qi Zhang, Zhiqing Xiao, Ruixuan Xiao, Lirong Gao, and Junbo Zhao. D. va: Validate your demonstration first before you use it. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2580–2594, 2025a.

Ziniu Zhang, Zhenshuo Zhang, Dongyue Li, Lu Wang, Jennifer Dy, and Hongyang R Zhang. Linear-time demonstration selection for in-context learning via gradient estimation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 16470–16488, 2025b.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pp. 12697–12706. Pmlr, 2021.

Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International conference on machine learning*, pp. 11492–11502. PMLR, 2020.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. VL-ICL bench: The devil in the details of multimodal in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cpGPPLLYYx.

APPENDIX CONTENTS

A Method and implementation details **18**

A.1 ICL classification . . . 18

A.2 ICL generation . . . 19

A.3 Skill-augmented reasoning . . . 19

B Experimental setup **20**

B.1 ICL classification . . . 20

B.2 ICL generation . . . 22

B.3 Skill-augmented reasoning . . . 23

B.4 Baselines . . . 24

C Additional results **26**

C.1 ICL classification . . . 26

C.2 Skill-augmented reasoning . . . 26

D Ablations **28**

D.1 Ordering pools and subset selection . . . 28

D.2 Score transformation and consensus aggregation . . . 29

E Analyses **31**

E.1 Model-dependent context utility . . . 31

E.2 Query-adaptive consensus decisions . . . 31

E.3 Effects of adding demonstrations across queries . . . 32

F Computational cost **33**

F.1 In-context learning . . . 33

F.2 Skill-augmented reasoning . . . 33

G Additional related work **33**

A METHOD AND IMPLEMENTATION DETAILS

This appendix details response scoring, sequence construction, and the implementation of CHORUS in each setting.

A.1 ICL CLASSIFICATION

Class scores and normalization. For a query q with demonstration sequence $\mathcal{D}_{\pi,k}(q)$, we score each candidate label $y_c \in \mathcal{Y}$ by its mean conditional token log-probability (Brown et al., 2020; Zhao et al., 2021),

$$\ell_{\pi,k}(c) = \frac{1}{|y_c|} \sum_{t=1}^{|y_c|} \log P_\theta(y_{c,t} \mid \mathcal{D}_{\pi,k}(q), q, y_{c,<t}) \,. \tag{18}$$

We average log-probabilities over label tokens only. Normalizing the exponentiated scores gives the class distribution $p_{\pi,k}$ of Section 4.1, which is standardized into $\tilde{p}_{\pi,k}$ by Eq. 9. The small constant ε stabilizes near-uniform vectors. Appendix D.2 compares alternative score transformations.

Sequence construction and orderings. The main method uses four orderings of the retrieved demonstrations: descending BERT cosine similarity (Relevance), ascending similarity (Reverse), and two random permutations with fixed seeds 0 and 1 (Shuffle). Table 7 also defines eight additional orderings used only in the expanded-pool ablations of Appendix D.

Table 7: Demonstration orderings used by CHORUS (Base) and the expanded-pool ablations (Bookend and Auxiliary).

Group	Ordering	Order of demonstrations in the prompt
Base (4)	Relevance	Most similar first. This is the retrieval order $\mathcal{C}_q$.
	Reverse	Least similar first.
	Shuffle (seeds 0, 1)	Random order, one permutation per seed.
Bookend (2)	Interleaved	Odd ranks from the start (1, 3, 5, ...), then even ranks toward the end (..., 6, 4, 2), so the two most similar demonstrations sit at the two ends.
	Block	The more similar half in descending similarity, then the less similar half in ascending similarity.
Auxiliary (6)	Shuffle (seeds 2–5)	Random order, one permutation per seed.
	Stride s ($s = 3, 7$)	Every s-th demonstration from rank 1, then from rank 2, and so on (for $s = 3$: 1, 4, 7, ..., 2, 5, 8, ...).

AG and CM implementation. Let K_q be the maximum number of demonstrations evaluated for query q. Classification evaluates every $k = 0, \ldots, K_q$. AG retains two of the four orderings ($m_{\mathrm{AG}} = 2$), and CM uses all four. We use a stability window of $W = 6$ positions, a CM margin threshold of $\tau = 0.05$, and weights $(w_{\mathrm{AG}}, w_{\mathrm{CM}}) = (1, 2)$. Selection and fallback follow Section 4.2. Ties between fallback positions are resolved in favor of the smallest k.

Prefix reuse. For ICL classification, successive prefixes within an ordering share all preceding demonstrations. We retain their attention keys and values in a KV-cache (Pope et al., 2023). Let $\mathbf{H}_\pi^{(k)}$ be the cache for the first k demonstrations. It is extended only with the next demonstration,

$$\mathbf{H}_\pi^{(k)} = \textsc{Forward}\Big(e_{\pi(k)} \mid \mathbf{H}_\pi^{(k-1)}\Big) \,, \tag{19}$$

where FORWARD returns the updated cache and $\mathbf{H}_\pi^{(0)}$ contains any fixed prompt prefix. For each evaluated value of k, the query and candidate labels are scored against this cache. Their temporary states are discarded, preserving the demonstration cache for the next prefix.

Without reuse, evaluating all prefixes repeatedly processes the same demonstrations, requiring $\mathcal{O}(K_q^2 \bar{L}_{\mathrm{demo}})$ demonstration tokens per ordering, where $\bar{L}_{\mathrm{demo}}$ is the mean demonstration length. Caching processes each demonstration once. Including query and label scoring gives

$$\mathcal{O}\big(K_q \bar{L}_{\mathrm{demo}} + (K_q + 1) L_{\mathrm{score}}\big) \tag{20}$$

newly processed tokens, where L_{score} counts query and label tokens processed to score all classes at one evaluated value of k. Attention still accesses the growing context, so this token accounting does not establish linear wall-clock time or FLOPs.

The orderings are evaluated sequentially, retaining $|\Pi|C(K_q + 1)$ class scores for consensus. The demonstration cache is released between orderings, although temporary label-scoring states also contribute to peak memory. Measured inference times are reported in Appendix F.1.

A.2 ICL GENERATION

Generation evidence. Here, k is the number of demonstrations in the prompt. Each prompt yields an answer with confidence weight $w = \exp(\ell)$, where ℓ is its mean generated-token log-probability. Each of the four base demonstration orderings defines one response sequence, ordered by increasing k. Equivalent answers contribute to the same candidate in Eq. 8, whose cumulative support is used by AG and CM.

Answer matching. Answers are matched after fixed normalizations. GeoQuery removes the answer wrapper, redundant outer parentheses, whitespace, and case differences. MTOP and SMCalFlow normalize whitespace while preserving case. Break normalizes first-line step markers, whitespace, and case. NL2Bash and WebQs normalize whitespace and case. These rules define agreement between responses, not the evaluation metrics. Matching nonempty answers from prompts with $k > 0$ contribute to the same candidate. We omit evaluated values of k for which all four orderings produce empty answers.

AG and CM implementation. AG retains two sequences and CM uses all four, with $\tau = 0.05$ and AG–CM weights $1 : 2$. We use $W = 2$ adjacent evaluated positions. The remaining selection rules follow Section 4.2.

A.3 SKILL-AUGMENTED REASONING

Prompt format and fixed reasoning plans. We use the benchmark's domain-specific system instruction and query template. Each user prompt has the following structure:

> `Relevant Skill:`
> [All annotated gold skill documents, concatenated]
>
> [Plan instruction, when applicable]
>
> [Benchmark query and solution template]

For each plan, the model continues from the assistant prefill shown below. Each solution is generated independently. The no-plan response uses neither an added instruction nor a prefill.

Answer matching and response scores. Responses with equivalent extracted final answers contribute to the same candidate y_c in Eq. 8. Answers match when they are equal ignoring case or equivalent under the domain's numeric or symbolic comparison.

The response score ℓ_b is the mean over the last $\min(64, L_b)$ generated-token log-probabilities, where L_b is the number available for response b. Its confidence weight is $w_b = \exp(\ell_b)$, or 1 when token scores are unavailable. Empty answers receive no votes, but their responses still count toward the budget.

Sequence construction and orderings. We first determine which responses are available at each budget. The no-plan response is added first, followed by plan-conditioned responses in one of six cyclic orders, so each plan occupies every plan position once. A budget of B includes the first B responses. We reuse the generated responses and average results over the six orders. For each available response set, we construct four response accumulation orders: forward, reverse, and two shuffles drawn sequentially from a random generator seeded with zero. These orders change how responses are accumulated for consensus, not which responses are generated.

AG and CM implementation. Both models use $(m_{\mathrm{AG}}, m_{\mathrm{CM}}, W, \tau) = (3, 4, 2, 0)$ with AG–CM weights $1 : 2$. AG retains three sequences ranked by stability, and CM uses all four. Here, W counts consecutive sequence positions, as in Section 4.2. These settings are fixed across domains, queries, runs, and response budgets. We return the first generated solution whose answer matches the selected candidate. Ties between candidates are resolved by first appearance.

Plan 1. formalize. Prefill: `## Setup`.

> Approach (formalize first): Begin with a section titled '## Setup' that lists every given quantity or premise as a named symbol or numbered fact, and states the exact target. Then derive the answer using only those symbols/facts, showing each transformation. Do not skip the Setup section.

Plan 2. enumerate. Prefill: `## Cases`.

> Approach (case enumeration): Begin with a section titled '## Cases' that exhaustively lists the distinct cases, candidates, or scenarios the problem admits. Resolve each case one by one, mark which ones are ruled out and why, and combine the surviving cases into the final answer.

Plan 3. skill_verbatim. Prefill: `## Step 1 (from skill).`

> Approach (follow the skill literally): Treat the Relevant Skill as a checklist. Walk through its procedure in order, writing '## Step i (from skill)' for each step and recording the intermediate value or conclusion at that step. Do not reorder or skip steps; only after the last step state the final answer.

Plan 4. alt_method. Prefill: `## Independent solution`.

> Approach (independent method, then cross-check): First solve the problem with a method that does NOT rely on the procedure in the Relevant Skill, under a section titled '## Independent solution'. Then, under '## Cross-check with skill', re-derive the answer using the skill's procedure. If the two disagree, find the error and state which one is correct.

Plan 5. estimate_first. Prefill: `## Estimate`.

> Approach (estimate, then compute): Begin with a section titled '## Estimate' giving a rough expected answer (order of magnitude, sign, range, or the most plausible option) with a one-line justification. Then compute the exact answer under '## Exact'. If the exact result falls outside the estimate, re-examine both before finalizing.

Plan 6. error_audit. Prefill: `## Pitfalls`.

> Approach (pitfalls first): Begin with a section titled '## Pitfalls' listing three specific mistakes that are commonly made on this type of problem (e.g., unit conversion, off-by-one, misreading a condition, a wrong formula). Then solve the problem, and finish with a '## Audit' section that checks the solution against each listed pitfall before giving the final answer.

B Experimental setup

Experiments are run on NVIDIA H200 and NVIDIA RTX A6000 GPUs.

B.1 ICL classification

B.1.1 Models and evaluation

We evaluate five models: Llama 3 8B and 70B (Grattafiori et al., 2024), Qwen2.5 7B Instruct (Qwen et al., 2025), Mistral 7B Instruct v0.3 (Jiang et al., 2023), and Gemma 2 27B (Team et al., 2024). Gemma 2 27B and Llama 3 70B are evaluated in bfloat16 precision. ICL prompts are capped at 6,144 tokens for all models. Label scoring follows Appendix A.1.

Each method predicts one label per query. Accuracy is the percentage of queries whose prediction matches the gold label, computed on the full validation split of each task in Table 8. Reported averages weight the 12 tasks equally.

B.1.2 DATASETS

We evaluate 12 classification tasks in three categories: **natural language understanding (NLU)**, **reasoning**, and **question answering (QA)**. MRPC, CoLA, and RTE are taken from GLUE (Wang et al., 2018), and WiC, CB, COPA, and WSC from SuperGLUE (Wang et al., 2019). Because these benchmarks do not release test labels, we follow CEIL (Ye et al., 2023) and use the original training splits for training and the labeled validation splits for evaluation. Table 8 lists each task with its training and evaluation (validation) split sizes and maximum number of demonstrations, and the descriptions below state what each task asks.

Table 8: ICL classification datasets. $K^{\max}$ is the task-specific upper limit on the requested number of demonstrations. The effective number also depends on the 6,144-token prompt limit.

Category	Dataset	Task	#Train	#Eval	$K^{\max}$
NLU	SST-5	Fine-grained sentiment analysis	8,534	1,101	200
	CoLA	Linguistic acceptability	8,528	1,043	275
	MRPC	Paraphrase detection	3,668	408	70
	WiC	Word sense disambiguation	5,428	638	100
	CB	Natural language inference	250	56	50
	RTE	Textual entailment	2,488	277	60
Reasoning	CommonsenseQA (CSQA)	Multiple-choice commonsense reasoning	9,740	1,221	225
	COPA	Causal reasoning	400	100	100
	WinoGrande (WinoG)	Commonsense pronoun resolution	40,394	1,267	125
	WSC	Commonsense coreference resolution	275	104	150
	PIQA	Physical commonsense reasoning	14,822	1,838	60
QA	OpenBookQA (OBQA)	Scientific question answering	4,831	500	100

Natural language understanding (NLU)

SST-5 (Socher et al., 2013). Movie-review sentences are labeled on a five-point sentiment scale: *very negative*, *negative*, *neutral*, *positive*, or *very positive*.

CoLA (Warstadt et al., 2019). The task decides whether a single sentence is grammatically acceptable.

MRPC (Dolan & Brockett, 2005). The task decides whether two sentences are semantically equivalent.

WiC (Pilehvar & Camacho-Collados, 2019). The task decides whether a word has the same meaning in two sentences.

CB (De Marneffe et al., 2019). Each premise and hypothesis pair is labeled as *entailment*, *contradiction*, or *neutral*.

RTE (Dagan et al., 2005). The task decides whether one text entails another. The data combine RTE1 (Dagan et al., 2005), RTE2 (Haim et al., 2006), RTE3 (Giampiccolo et al., 2007), and RTE5 (Bentivogli et al., 2009). SuperGLUE uses the same data, and we use the GLUE split.

Reasoning

CommonsenseQA (Talmor et al., 2019). Each question has five answer options and requires commonsense knowledge.

COPA (Roemmele et al., 2011). Given a premise, the model selects the more plausible cause or effect from two choices.

WinoGrande (Sakaguchi et al., 2020). The model fills a blank in a sentence with the appropriate candidate. The benchmark extends WSC to a larger dataset with reduced dataset bias.

WSC (Levesque et al., 2012). Given a sentence, a pronoun, and a candidate span, the task decides whether the pronoun refers to the span.

PIQA (Bisk et al., 2020). The model chooses the better of two candidate solutions to a problem that requires physical commonsense.

Question answering (QA)

OpenBookQA (Mihaylov et al., 2018). Four-choice questions test elementary-level science knowledge and reasoning. The dataset provides about 1.3k open-book facts, which we exclude.

B.2 ICL GENERATION

B.2.1 MODELS AND EVALUATION

The generation evaluation uses Llama 3 8B (Grattafiori et al., 2024) with greedy decoding and a 6,144-token prompt limit, at which prompts are truncated.

Each task uses the metric of its benchmark implementation. GeoQuery, MTOP, and SMCalFlow use exact match between the extracted prediction and the gold logical form (FunQL, intent and slot parse, and Lispress, respectively). WebQs counts a prediction as correct if it matches any gold answer after lowercasing and removing punctuation and articles. NL2Bash uses corpus-level 4-gram BLEU against the reference command. Break uses logical-form exact match (LF-EM), which converts the predicted and gold decompositions into logical forms and compares their structure. Scores are reported on a 0–100 scale, and averages weight the six tasks equally. The evaluation covers structured and short-form generation.

B.2.2 DATASETS

We evaluate six generation tasks in two categories: **semantic parsing (SP)** and **question understanding and answering (QUA)**. Demonstrations are retrieved from each task's training split, and all methods are evaluated on the same 200 held-out queries per task. Table 9 lists each task with its split sizes and maximum number of demonstrations, and the descriptions below state what each task asks.

Table 9: ICL generation datasets. $K^{\max}$ is the task-specific upper limit on the requested number of demonstrations.

Category	Dataset	Task	#Train	#Held-out	#Eval	$K^{\max}$
SP	GeoQuery	Semantic parsing (FunQL)	600	280	200	277
	MTOP	Task-oriented semantic parsing	15,667	2,235	200	176
	NL2Bash	Natural language to Bash	8,090	609	200	179
	SMCalFlow	Dialogue to dataflow program	133,584	14,751	200	105
QUA	Break	Question decomposition (QDMR)	44,321	7,760	200	104
	WebQs	Open-domain question answering	3,778	2,032	200	300

Semantic parsing (SP)

GeoQuery (Zelle & Mooney, 1996). Questions about U.S. geography are mapped to executable FunQL logical forms over geographic entities, relations, and attributes.

MTOP (Li et al., 2021). User utterances are mapped to hierarchical intent and slot representations, which can be nested for compositional requests.

NL2Bash (Lin et al., 2018). Descriptions of command-line operations are mapped to executable Bash commands with the appropriate utilities, flags, and arguments.

SMCalFlow (Andreas et al., 2020). A dialogue turn is mapped to an executable dataflow program that represents the user's intent and can refer to or revise information from earlier turns.

Question understanding and answering (QUA)

Break (Wolfson et al., 2020). A complex question is decomposed into an ordered sequence of simpler steps in Question Decomposition Meaning Representation (QDMR).

WebQs (Berant et al., 2013). Factoid questions, mostly about named entities, have answers grounded in Freebase. A question can have several valid answers.

B.3 SKILL-AUGMENTED REASONING

B.3.1 MODELS AND EVALUATION

We evaluate Qwen3-4B and Qwen3-8B (Yang et al., 2025a) with thinking disabled. CHORUS responses and single-response greedy baselines use greedy decoding, while Self-Consistency and sampling baselines use temperature 0.7. Each generation request allows up to 4,096 output tokens.

Each domain uses its SRA-Bench evaluator, which extracts a final answer from the full solution and compares it with the reference. TheoremQA accepts numeric answers within a 4% relative tolerance. CHAMP accepts the same numeric tolerance, requires integer answers to match after rounding, and also accepts canonical-string and symbolic equivalence. MedCalc-Bench requires decimal answers to fall within the reference lower and upper limits, integer answers to match after rounding, and date and gestational-age answers to match exactly. LogicBench requires the exact yes/no or answer-choice label. Parser state is reset between answers, and all methods are scored with latex2sympy2 1.5.4 and SymPy 1.14.0.

Accuracy is the percentage of correct questions in each domain, and averages weight the four domains equally. We report the mean and standard deviation over three inference runs on the same questions.

B.3.2 DATASETS

We evaluate 1,546 questions from four SRA-Bench (Su et al., 2026) domains, built from the datasets in Table 10. Every method receives the same annotated gold skill documents for each question, including all documents when multiple skills are annotated. This setting evaluates how models use supplied skills rather than how the skills are retrieved. The descriptions below state what each domain asks and what its skill documents supply.

Table 10: Skill-augmented reasoning datasets, with the number of evaluation questions and distinct gold skills in each domain.

Dataset	Task	#Questions	#Gold skills
TheoremQA	Theorem application	482	320
LogicBench	Logical reasoning	380	19
MedCalc-Bench	Medical calculation	550	55
CHAMP	Mathematical reasoning	134	85
Total	–	1,546	479

TheoremQA (Chen et al., 2023). Questions require applying mathematical and scientific theorems, and the skill documents give the relevant theorem or procedure.

LogicBench (Parmar et al., 2024). Logical reasoning problems follow reusable reasoning rules and patterns, which the skill documents specify.

MedCalc-Bench (Khandekar et al., 2024). Medical calculation problems apply clinical calculators to the information in the query, and the skill documents define each calculator and how to apply it.

CHAMP (Mao et al., 2024). Mathematical problems can involve several annotated skills, so the model applies the supplied concepts and procedures during multi-step reasoning.

B.4 BASELINES

B.4.1 ICL DEMONSTRATION SELECTION

We compare CHORUS with 12 demonstration selection baselines across both ICL classification and ICL generation. Ten select examples or their ordering at a fixed number of demonstrations, spanning random and similarity-based retrieval, diversity-aware selection, and LLM-guided optimization. DR-IKE and Adaptive-k are adapted to choose the number of demonstrations for each query from their original settings of in-context knowledge editing and passage selection in long-context QA, respectively. Table 11 lists the number of demonstrations requested by each baseline. Table 5 separately varies the requested number of demonstrations for three baselines. Increasing this externally set number is distinct from learning or computing a query-specific stopping decision.

Table 11: Number of demonstrations requested by each ICL baseline. The prompt limit can reduce the effective number.

Requested k	**Baselines**
50	RANDOM, TOPK-BM25, TOPK-BERT, DPP-BERT, EPR
8	D.VA, TOPICK
5	EASE
3	SE2, SEDPO
Chosen per query	DR-IKE, ADAPTIVE-k

Random and similarity-based retrieval

RANDOM. A retriever that randomly selects samples from the training set without replacement, independent of the test query.

TOPK-BM25. BM25 (Robertson & Zaragoza, 2009) is a retrieval method that extends TF-IDF, computing lexical similarity scores between a test query and demonstration samples, and constructing the demonstration set using the top-k highest-scoring samples.

TOPK-BERT. A dense retriever that selects the top-k samples based on the cosine similarity between the BERT embeddings (Devlin et al., 2019) of the test query and each sample in the candidate demonstration pool.

Diversity- and coverage-aware selection

DPP-BERT. A dense retriever that selects a diverse set of samples by applying DPP-based MAP inference (Chen et al., 2018) on BERT embedding similarity.

TOPICK (Kweon et al., 2025). A topic coverage-based retriever that samples demonstrations by jointly considering the topics required by the test query, the topics covered by each demonstration candidate, and the LLM's inherent topical knowledge. It trains a lightweight predictor to estimate these factors and iteratively selects examples that cover the required topics of the test query while exhibiting low topical knowledge in the LLM.

LLM-guided optimization

D.VA (Zhang et al., 2025a). A self-adaptive demonstration selection method (Wu et al., 2023; Peng et al., 2024). From a candidate set, D.Va takes the sample most similar to the test input as a validation example, scores each remaining demonstration by the perplexity of a prompt constructed with this validation example, and selects the top k demonstrations.

EPR (Rubin et al., 2022). A learning-based retriever trained from LLM scores. For each query, 50 demonstrations are first retrieved using BM25 (Robertson & Zaragoza, 2009), and the LLM score of each demonstration used alone is computed. BERT embeddings (Devlin et al., 2019) are then trained with a contrastive loss. At inference, the trained encoder scores each sample by its inner product with the query, and the top-k samples are selected.

EASE (Wu et al., 2024). A retriever that selects one ordered demonstration sequence for the whole task, independent of the test query. To reduce the large space of candidate sequences, EASE first filters out task-irrelevant candidates by their feature-space distance to the validation set. It then uses NeuralUCB (Zhou et al., 2020) to choose new sequences to evaluate, conditioned on those evaluated so far, and returns the sequence with the highest LLM score within the evaluation budget.

SE2 (Liu et al., 2024). A sequential retriever trained from LLM feedback under progressively extended demonstration contexts. During supervision collection, it samples candidate examples and scores them using the normalized likelihood of the correct label for classification or a task-specific output metric for generation. Candidates are sampled according to their score ranks to extend the context and collect feedback for the next step. The retriever uses the highest-scoring candidate as a positive example in an InfoNCE (Oord et al., 2018) objective. At inference, beam search constructs demonstration sequences using the learned retrieval scores.

SEDPO (Ji et al., 2025). A retriever trained with DPO to learn which examples are relatively more useful for a given test query, rather than classifying each demonstration as good or bad. Rankings derived from the LLM performance of each demonstration sequence serve as teacher signals, and DPO aligns the retriever with these rankings so that it can better select the next demonstration.

Query-adaptive number of demonstrations

DR-IKE (Nafee et al., 2025). Originally designed for in-context knowledge editing, DR-IKE adds demonstrations sequentially and stops using a learned threshold σ. During training, the task-specific threshold is raised to the softmax probability at which predictions change from correct to incorrect.
In our adaptation without the fixed editing examples, the raw threshold led to stopping at $k = 1$. We instead store the ratio of the transition probability to the top-1 probability and use σ times the current top-1 probability as the inference threshold. We report DR-IKE with this relative-confidence threshold.

ADAPTIVE-k (Taguchi et al., 2025). A sample-number adaptive retrieval method for retrieval-augmented generation (RAG) that sets the number of retrieved documents from the test query alone, without iterative LLM calls or additional training. It sorts candidates by cosine similarity to the query, computes the first differences of the sorted scores, and selects candidates up to the largest similarity gap, which the original work observes as a sharp drop on RAG benchmarks.

B.4.2 SKILL-AUGMENTED REASONING

Self-Consistency (SC) and Adaptive-Consistency (AC) share six sampled responses per question from the same prompt, retaining the gold skill documents and domain instructions but omitting the plan instruction and prefill. Empty answers consume the response budget but contribute no votes. Ties are resolved by first appearance.

Full-Skill Injection (Su et al., 2026). The method inserts the supplied gold documents without an added plan and generates one response, with greedy decoding or sampling.

Self-Consistency (Wang et al., 2023b). SC-B uses the first B responses. SC (majority) selects the most frequent answer and returns its first solution. SC (weighted) selects the answer with the largest sum of response confidence weights w_b (Wang et al., 2023b) and returns its highest-scoring solution.

Adaptive-Consistency (Aggarwal et al., 2023). AC checks responses sequentially and stops when the most frequent answer has sufficient support over its runner-up. After response n, let a_n and b_n be their observed counts, with $b_n = 0$ when only one answer has appeared. The Beta stopping criterion is

$$\gamma_n = \Pr(Z_n > 1/2), \qquad Z_n \sim \mathrm{Beta}(a_n + 1, b_n + 1). \tag{21}$$

AC stops at the first $\gamma_n \geq 0.95$ or after six responses and returns the majority answer's first solution among the responses observed so far. While all observed answers are empty, AC continues sampling up to the cap. With a six-response cap, early stopping requires at least four matching answers, so the remaining responses cannot change the majority.

C ADDITIONAL RESULTS

C.1 ICL CLASSIFICATION

Tables 12, 13, 14, and 15 extend the 12-task classification comparison to four additional models. Baselines are cited in Table 2 and described in Appendix B.4. Bold marks the highest accuracy in each column, including ties.

CHORUS achieves the highest average accuracy on all four models, exceeding the strongest baseline by 2.15, 2.32, 2.05, and 0.17 percentage points on Qwen2.5 7B Instruct, Mistral 7B Instruct v0.3, Gemma 2 27B, and Llama 3 70B, respectively. The leading method on individual tasks varies across tasks and models. These results extend the average-performance advantage observed on Llama 3 8B while showing that the relative strengths of context-use methods depend on the task and target model.

Table 12: ICL classification performance on Qwen2.5 7B Instruct across 12 tasks. Accuracy is reported in percent. Avg. is the mean accuracy across the 12 tasks.

Method	SST-5	CoLA	MRPC	WiC	CSQA	COPA	CB	WinoG	WSC	OBQA	PIQA	RTE	Avg.
RANDOM	51.23	58.79	72.79	71.47	75.68	99.00	85.71	67.80	63.46	86.20	87.32	46.21	72.14
TOPK-BM25	53.59	63.15	72.30	72.26	74.77	99.00	83.93	68.90	68.27	87.00	88.41	51.62	73.60
TOPK-BERT	50.77	64.77	**74.02**	70.85	76.00	**100.00**	**91.07**	69.06	62.50	**88.60**	87.65	52.71	74.00
DPP-BERT	51.86	63.55	73.77	72.57	**76.82**	**100.00**	89.29	69.38	64.42	**88.60**	87.81	54.51	74.38
D.VA	49.59	61.76	72.30	69.12	75.59	99.00	89.29	**70.09**	69.23	**88.60**	88.30	52.35	73.77
EPR	**55.13**	64.92	73.77	72.41	75.18	99.00	85.71	68.03	69.23	86.60	88.68	52.71	74.28
TOPICK	48.96	62.47	70.59	67.08	72.81	97.00	85.71	67.09	67.31	87.40	87.11	53.43	72.25
EASE	49.95	61.52	67.40	69.12	70.43	98.00	87.50	69.85	68.27	84.80	86.51	47.29	71.72
SE2	45.96	**83.80**	73.04	62.54	71.17	97.00	82.14	67.25	67.31	86.00	86.83	**57.40**	73.37
SEDPO	44.23	82.45	73.53	63.32	70.76	98.00	89.29	67.64	63.46	85.80	86.78	52.71	73.16
DR-IKE	37.60	59.97	71.08	62.54	70.43	97.00	83.93	66.61	64.42	82.80	85.96	51.26	69.47
ADAPTIVE-k	38.69	83.03	71.81	60.50	71.25	98.00	83.93	66.22	65.38	83.80	85.85	50.18	71.55
CHORUS	53.04	82.93	73.77	**73.20**	76.41	**100.00**	**91.07**	69.30	**71.15**	86.00	**88.74**	52.71	**76.53**

Table 13: ICL classification performance on Mistral 7B Instruct v0.3 across 12 tasks. Accuracy is reported in percent. Avg. is the mean accuracy across the 12 tasks.

Method	SST-5	CoLA	MRPC	WiC	CSQA	COPA	CB	WinoG	WSC	OBQA	PIQA	RTE	Avg.
RANDOM	52.13	50.22	72.55	64.89	74.77	94.00	92.86	62.75	50.00	81.00	83.84	47.29	68.86
TOPK-BM25	52.77	52.70	74.26	65.67	75.76	97.00	**96.43**	62.12	**52.88**	81.00	83.03	47.29	70.08
TOPK-BERT	52.13	55.41	73.28	65.67	76.25	**98.00**	94.64	62.27	47.12	83.80	82.64	47.29	69.88
DPP-BERT	53.95	56.11	73.53	64.11	**76.99**	**98.00**	91.07	62.90	47.12	**84.20**	83.57	47.29	69.90
D.VA	49.23	55.72	72.55	65.05	76.74	**98.00**	91.07	65.90	48.08	84.00	**85.47**	48.74	70.05
EPR	52.68	53.09	**77.94**	**66.93**	76.17	96.00	94.64	62.27	50.00	80.80	83.84	47.29	70.14
TOPICK	49.95	56.77	74.02	62.38	75.02	94.00	89.29	64.40	41.35	83.80	85.04	47.29	68.61
EASE	48.59	52.53	73.28	65.20	70.35	93.00	94.64	61.64	51.92	79.20	85.36	47.29	68.58
SE2	40.33	80.25	72.55	62.07	73.38	97.00	89.29	**66.54**	50.96	81.20	83.84	50.90	70.69
SEDPO	41.78	**80.92**	71.32	64.11	71.99	94.00	89.29	64.96	45.19	81.80	84.22	48.74	69.86
DR-IKE	33.79	50.62	73.04	63.17	72.56	**98.00**	82.14	63.38	50.96	78.40	84.06	51.99	66.84
ADAPTIVE-k	37.97	79.96	71.57	61.91	72.73	96.00	89.29	64.64	47.12	80.60	84.39	51.26	69.79
CHORUS	**54.41**	80.54	73.53	65.67	76.90	97.00	**96.43**	63.85	48.08	83.00	83.95	**52.71**	**73.01**

C.2 SKILL-AUGMENTED REASONING

The main text reports CHORUS with six responses, matching SC's fixed six-response budget and AC's maximum of six responses. Table 16 reports domain-level results over three runs at this budget and at seven responses, which include the no-plan response and all six plans.

Relative to SC-6 (weighted), the strongest SC or AC baseline on average, improvements are most pronounced on TheoremQA for Qwen3-4B and on CHAMP and TheoremQA for Qwen3-8B. The gains vary across domains, with a small decrease on LogicBench for Qwen3-8B. Adding the seventh response increases mean accuracy on both models.

Table 14: ICL classification performance on Gemma 2 27B across 12 tasks. Accuracy is reported in percent. Avg. is the mean accuracy across the 12 tasks.

Method	SST-5	CoLA	MRPC	WiC	CSQA	COPA	CB	WinoG	WSC	OBQA	PIQA	RTE	Avg.
RANDOM	50.32	61.59	72.30	68.34	78.54	98.00	94.64	76.64	72.12	89.60	86.51	47.29	74.66
TOPK-BM25	53.13	62.35	75.49	69.44	76.25	99.00	91.07	**78.22**	72.12	90.40	87.05	47.29	75.15
TOPK-BERT	53.22	68.59	73.77	66.93	**79.03**	**100.00**	91.07	75.14	74.04	**92.00**	86.07	47.29	75.60
DPP-BERT	**54.04**	67.56	74.51	68.03	78.13	**100.00**	94.64	74.35	61.54	91.80	86.78	47.29	74.89
D.VA	50.41	61.36	72.06	68.81	77.15	99.00	85.71	76.56	72.12	90.60	**87.27**	47.29	74.03
EPR	53.95	63.80	80.39	70.38	76.00	99.00	**98.21**	76.87	65.38	90.80	86.83	47.29	75.74
TOPICK	50.59	55.34	71.81	67.40	77.15	99.00	85.71	75.22	65.38	91.20	86.56	47.29	72.72
EASE	50.05	55.22	68.38	**71.79**	77.97	99.00	85.71	76.56	73.08	89.80	**87.27**	47.29	73.51
SE2	43.87	62.41	73.28	66.93	73.55	99.00	80.36	76.72	71.15	89.20	84.93	47.65	72.42
SEDPO	47.59	66.48	**80.64**	67.71	73.96	99.00	78.57	76.16	**75.00**	90.20	86.02	47.29	74.05
DR-IKE	36.42	52.10	67.40	63.01	73.22	99.00	71.43	75.06	61.54	89.80	83.90	43.32	68.02
ADAPTIVE-k	38.06	83.80	68.87	63.48	70.27	99.00	76.79	73.48	67.31	89.60	84.11	49.82	72.05
CHORUS	52.77	**86.00**	76.96	69.59	**79.03**	99.00	89.29	76.24	73.08	91.60	87.21	**52.71**	**77.79**

Table 15: ICL classification performance on Llama 3 70B across 12 tasks. Accuracy is reported in percent. Avg. is the mean accuracy across the 12 tasks.

Method	SST-5	CoLA	MRPC	WiC	CSQA	COPA	CB	WinoG	WSC	OBQA	PIQA	RTE	Avg.
RANDOM	50.59	84.85	73.77	70.22	78.95	99.00	92.86	81.45	83.65	92.80	87.27	47.29	78.56
TOPK-BM25	51.86	85.23	74.75	71.94	78.62	99.00	94.64	**85.24**	85.58	**94.80**	88.19	47.29	79.76
TOPK-BERT	52.04	86.10	75.00	71.79	79.85	99.00	94.64	81.61	81.73	94.60	88.41	47.29	79.34
DPP-BERT	52.13	87.15	77.21	72.57	**80.84**	99.00	96.43	81.61	**86.54**	94.60	88.47	47.29	80.32
D.VA	47.50	85.62	72.30	71.94	77.15	99.00	91.07	81.85	83.65	94.40	88.85	48.01	78.45
EPR	**56.22**	83.80	80.88	**73.98**	76.58	99.00	**98.21**	80.90	78.85	94.60	88.25	47.29	79.88
TOPICK	51.04	83.41	71.32	70.38	77.15	99.00	89.29	80.98	83.65	92.40	**90.21**	48.01	78.07
EASE	46.68	84.37	67.16	66.14	75.51	99.00	89.29	78.22	80.77	89.60	87.81	47.29	75.99
SE2	44.32	84.37	68.14	71.47	74.20	**100.00**	87.50	81.06	74.04	91.00	89.28	45.85	75.94
SEDPO	46.41	83.03	**82.84**	72.57	72.97	**100.00**	75.00	81.85	77.88	90.20	89.17	50.90	76.90
DR-IKE	35.15	82.07	69.12	69.91	72.24	99.00	83.93	77.27	76.92	90.00	83.46	49.46	74.04
ADAPTIVE-k	37.06	83.70	66.67	66.30	70.84	99.00	83.93	78.53	76.92	89.80	87.27	52.35	74.36
CHORUS	52.13	**87.54**	76.23	73.67	79.93	99.00	94.64	83.50	82.69	94.60	89.23	**52.71**	**80.49**

Baselines and SC-B follow Appendix B.4.2. SC-4 matches the four-response points in Figure 2. At one response, CHORUS uses the no-plan greedy solution while SC uses sampling, so their first points differ before any consensus.

Model-dependent effects of reasoning plans. Table 17 shows that the benefit of a reasoning plan depends on the target model. For example, skill_verbatim underperforms the no-plan baseline on Qwen3-4B but achieves the highest single-plan accuracy on Qwen3-8B. This variation motivates combining responses from multiple plans rather than relying on one plan across models. The best single-plan reference uses the plan with the highest evaluation accuracy averaged across domains and runs for each model, held fixed across queries.

Table 16: Skill-augmented reasoning accuracy by domain. Values are means ± SD over three runs, with Avg. weighting domains equally.

Method	Decoding	Aggregation	Response budget	TheoremQA	LogicBench	MedCalc	CHAMP	Avg.
Qwen3-4B								
Best single plan Full-Skill Injection (gold context, no plan)	Greedy	–	1	67.7 ± 0.1	87.5 ± 0.8	73.5 ± 0.8	75.1 ± 2.2	75.96 ± 0.37
Full-Skill Injection (gold context, no plan)	Greedy	–	1	68.5 ± 0.3	87.2 ± 0.5	67.9 ± 0.5	71.1 ± 1.7	73.70 ± 0.62
Full-Skill Injection (gold context, no plan)	Sampling	–	1	68.6 ± 0.5	86.4 ± 1.2	69.9 ± 1.8	72.4 ± 2.0	74.33 ± 0.16
Self-Consistency	Sampling	Weighted	4	70.3 ± 0.1	88.8 ± 0.8	71.3 ± 0.8	78.1 ± 0.9	77.12 ± 0.57
Self-Consistency	Sampling	Majority	6	70.7 ± 0.4	88.6 ± 0.5	71.9 ± 0.9	78.1 ± 0.4	77.33 ± 0.34
Self-Consistency	Sampling	Weighted	6	71.1 ± 0.2	88.9 ± 1.1	71.9 ± 0.9	78.4 ± 1.3	77.55 ± 0.21
Adaptive-Consistency	Sampling	Majority	6	70.7 ± 0.4	88.6 ± 0.5	71.9 ± 0.9	78.1 ± 0.4	77.33 ± 0.34
CHORUS	Greedy	AG+CM	6	73.5 ± 0.4	90.5 ± 0.4	72.1 ± 0.1	79.9 ± 0.7	78.98 ± 0.31
CHORUS	Greedy	AG+CM	7	73.7 ± 0.6	91.1 ± 0.1	73.2 ± 0.3	79.9 ± 0.8	79.46 ± 0.14
Qwen3-8B								
Best single plan Full-Skill Injection (gold context, no plan)	Greedy	–	1	69.0 ± 0.3	88.4 ± 0.5	75.0 ± 0.2	73.1 ± 0.0	76.39 ± 0.10
Full-Skill Injection (gold context, no plan)	Greedy	–	1	61.3 ± 0.5	86.8 ± 1.9	76.0 ± 0.5	58.5 ± 1.1	70.64 ± 0.59
Full-Skill Injection (gold context, no plan)	Sampling	–	1	60.7 ± 1.1	85.4 ± 1.2	74.6 ± 0.7	67.7 ± 4.5	72.08 ± 1.14
Self-Consistency	Sampling	Weighted	4	64.9 ± 1.5	89.8 ± 0.8	76.2 ± 1.0	71.4 ± 1.9	75.58 ± 0.08
Self-Consistency	Sampling	Majority	6	65.8 ± 0.8	89.7 ± 0.3	76.1 ± 0.6	73.1 ± 1.3	76.21 ± 0.36
Self-Consistency	Sampling	Weighted	6	66.5 ± 0.7	90.3 ± 0.5	76.1 ± 0.6	72.9 ± 2.4	76.43 ± 0.69
Adaptive-Consistency	Sampling	Majority	6	65.8 ± 0.8	89.7 ± 0.3	76.1 ± 0.6	73.1 ± 1.3	76.21 ± 0.36
CHORUS	Greedy	AG+CM	6	70.9 ± 0.3	89.9 ± 0.4	78.2 ± 0.3	79.8 ± 0.3	79.69 ± 0.17
CHORUS	Greedy	AG+CM	7	71.2 ± 0.4	90.1 ± 0.9	78.7 ± 0.3	80.4 ± 0.5	80.12 ± 0.32

Table 17: Accuracy of individual reasoning plans and the no-plan baseline, averaged across four domains. Values are means ± SD over three runs. Bold identifies the best single plan for each model.

Plan	Qwen3-4B	Qwen3-8B
Full-Skill Injection (gold context)	73.70 ± 0.62	70.64 ± 0.59
formalize	74.35 ± 0.24	75.70 ± 1.13
enumerate	73.22 ± 0.46	72.97 ± 0.31
skill_verbatim	71.47 ± 0.42	$\mathbf{76.39 \pm 0.10}$
alt_method	66.00 ± 0.65	74.62 ± 0.33
estimate_first	62.05 ± 0.39	72.82 ± 0.33
error_audit	$\mathbf{75.96 \pm 0.37}$	75.97 ± 0.27

D ABLATIONS

We examine how ordering selection, score transformation, and aggregation parameters affect consensus. The ICL analyses use the four base orderings or expanded pools drawn from the 12 orderings in Appendix A.1; each comparison specifies its pool. ICL accuracies are averaged over 12 tasks and reported as percentages. We also test aggregation sensitivity in skill-augmented reasoning.

D.1 ORDERING POOLS AND SUBSET SELECTION

Ordering pool. Table 18(a) expands the four base orderings with bookend and auxiliary orderings while retaining AG+CM. More orderings do not consistently improve accuracy. The full 12-ordering pool gives a modest improvement at three times the sequence evaluations, supporting the four base orderings as a balance between accuracy and evaluation cost.

Signals for selecting subsets. For ICL classification, the four-ordering AG rule selects two orderings by stability s_{stab} (Appendix A.1), while CM uses all four base orderings. For ICL ablations that retain fewer orderings than the available pool, we rank orderings for CM by early–late consistency,

$$s_{\mathrm{elc}}(\pi, q) = \mathbb{1}\big[\mathrm{maj}(\hat{c}_\pi^{\mathrm{early}}) = \mathrm{maj}(\hat{c}_\pi^{\mathrm{late}})\big]. \tag{22}$$

Here, $\hat{c}_\pi^{\text{early}}$ contains the top-class predictions at the first $\lfloor 0.4n \rfloor$ valid positions, and $\hat{c}_\pi^{\text{late}}$ contains the remaining predictions, where n is the number of evaluated positions, including $k = 0$. The majority function breaks ties by smallest class index. Queries with fewer than four valid positions receive a score of 0.5. CM retains the top m_{CM} orderings under this signal and applies its maximum aggregation to that subset. The pool comparison fixes $m_{\text{CM}} = 4$, while the subset-size ablation varies it. This ranking is only needed when selecting a subset; the main ICL CM rule uses all four base orderings.

Subset size. Table 18(b) fixes the four base orderings and varies how many enter AG and CM, complementing the candidate-pool comparison in panel (a). Both panels use the main AG+CM normalization, stable evidence selection, and $1:2$ weights. AG ranks orderings by s_{stab} and CM by early–late consistency.

Table 18: Effect of ordering selection on AG+CM accuracy (%) on Llama 3 8B, averaged over 12 tasks. Panel (a) varies the candidate pool, and panel (b) varies the number retained from the four base orderings. Bold marks the best accuracy in each panel, and $\star$ marks the reported setting.

(a) Candidate ordering pool

Pool	$\lvert\Pi\rvert$	Accuracy
Four base orderings $\star$	4	71.92
+ bookend	6	72.16
+ auxiliary	10	71.41
+ bookend and auxiliary	12	**72.19**

(b) Retained orderings ($\lvert\Pi\rvert = 4$)

	m_{CM}			
m_{AG}	1	2	3	4
1	71.04	71.07	71.50	71.79
2	71.01	71.17	71.82	**71.92** $\star$
3	71.06	71.21	71.35	71.54
4	71.09	71.10	71.48	71.32

Increasing m_{CM} from 1 to 4 raises accuracy for every m_{AG}, whereas with $m_{\text{CM}} = 4$ accuracy peaks at $m_{\text{AG}} = 2$ and falls to 71.32% when AG also uses all four orderings. CM benefits from more orderings because its maximum retains strong support from any of them, while AG benefits from excluding less stable orderings from its geometric mean. This matches the reported setting, in which AG retains two orderings and CM uses all four. As a separate reference without consensus, uniformly sampling one ordering and one of its evaluated numbers of demonstrations, including $k = 0$, gives an exact expected accuracy of 68.92%, 3.00 points below CHORUS.

Selected orderings. Table 19 summarizes the selected orderings as 12-task averages, using the larger $\lvert\Pi\rvert$=12 pool in our ablations. The two signals draw very different patterns from the same pool: s_{stab} selects relevance and the first shuffle most often, whereas s_{elc} spreads its selections roughly uniformly across the four base orderings. These frequencies describe the subset rules rather than establish an intrinsic ranking of orderings: early–late consistency is a binary score for queries with at least four valid positions, so multiple orderings can tie.

Table 19: Ordering-selection frequencies on Llama 3 8B with $\lvert\Pi\rvert = 12$. Results are averaged over 12 tasks, and bold marks the highest frequency for each signal, including ties.

Ordering	s_{stab}	s_{elc}
Relevance	**0.300**	**0.224**
Shuffle (seed 0)	0.274	0.221
Shuffle (seed 1)	0.107	**0.224**
Reverse	0.069	0.219
Bookend, interleaved	0.046	0.050
Bookend, block	0.031	0.024

Ordering	s_{stab}	s_{elc}
Shuffle (seed 2)	0.041	0.013
Shuffle (seed 3)	0.044	0.011
Shuffle (seed 4)	0.025	0.006
Shuffle (seed 5)	0.023	0.004
Stride 3	0.021	0.002
Stride 7	0.021	0.002

D.2 SCORE TRANSFORMATION AND CONSENSUS AGGREGATION

Normalized answer scores. Table 20 compares the standardization followed by softmax of Eq. 9 with alternative transformations under the same AG+CM procedure: the original class probabilities, log-odds, and probabilities smoothed over sequence positions by a moving average (window size 3

or 5) or a cumulative mean. Standardization followed by softmax gives the best accuracy with both ordering pools.

Table 20: AG+CM accuracy (%) under different class-score transformations on Llama 3 8B, averaged over 12 tasks. Bold marks the best accuracy in each column, and ⋆ marks the reported transformation.

Transformation	$\lvert\Pi\rvert$=12	$\lvert\Pi\rvert$=4
Original probabilities	71.24	71.68
Standardize + softmax ⋆	**72.19**	**71.92**
Log-odds	71.41	71.50
Moving average (3)	71.10	71.11
Moving average (5)	71.00	71.00
Cumulative mean	71.16	71.03

Ensemble weights. Table 21(a) varies the relative AG–CM weights within the complete CHORUS, keeping ordering selection and stable evidence selection fixed. Giving CM greater weight improves on equal weighting, but the optimum is not narrow: tested CM-to-AG ratios from 2 to 10 remain within 0.05 percentage points of the best accuracy. This supports the shared $1:2$ weighting without requiring a precisely tuned ratio.

Margin threshold. Table 21(b) varies the required margin τ between CM's two highest answer scores. Standalone CM uses all four orderings without stability selection, taking the latest position satisfying the margin and falling back to the latest valid position. CHORUS retains its ordering and stability selection. It remains more accurate throughout the sweep and performs near its best across several thresholds, including the reported $\tau = 0.05$ shared across ICL tasks.

Table 21: Accuracy (%) across (a) AG–CM weights and (b) CM margin thresholds on Llama 3 8B, averaged over 12 tasks with four base orderings. Bold marks the best accuracy in each column, and ⋆ marks the reported settings.

(a) AG–CM weights

$(w_{\mathrm{AG}}, w_{\mathrm{CM}})$	12-task Avg.
$(1,1)$	71.45
$(1,2)$ ⋆	**71.92**
$(1,3)$	71.87
$(1,4)$	71.89
$(1,5)$	71.89
$(1,10)$	71.89

(b) CM margin threshold

τ	CM	CHORUS
0.00	71.05	71.67
0.05 ⋆	71.03	71.92
0.10	71.05	**71.99**
0.20	**71.20**	71.83
0.50	70.43	71.70
0.60	69.95	71.70
0.70	71.06	71.87
0.80	70.29	71.34
0.90	70.29	71.34

Skill aggregation sensitivity. At a six-response budget, we test 1,200 combinations per model: $W \in \{2,3,4\}$, $m_{\mathrm{AG}}, m_{\mathrm{CM}} \in \{1,2,3,4\}$, $\tau \in \{0, 0.025, 0.05, 0.1, 0.2\}$, and CM-to-AG weight ratios in $\{0.25, 0.5, 1, 2, 4\}$. AG subsets are ranked by stability and CM subsets by early–late consistency, applied to response-accumulation positions. Results average six cyclic plan-addition orders and three runs, with equal weights across four domains. Table 22 shows that even the lowest accuracy exceeds both Self-Consistency variants and Adaptive-Consistency on each model. This advantage holds across the tested grid rather than requiring a narrowly tuned combination.

Table 22: Skill accuracy across 1,200 aggregation settings per model at a six-response budget. Values average four domains and three runs.

Model	CHORUS across settings			Self-Consistency (Sampling, Weighted) 6 responses
	Min.	Median	Max.	
Qwen3-4B	78.29	78.85	79.01	77.55
Qwen3-8B	78.78	79.65	79.80	76.43

E ANALYSES

E.1 MODEL-DEPENDENT CONTEXT UTILITY

Scope of utility supervision. Table 1 describes the exhaustive ICL candidate space. Practical selectors obtain supervision from a subset of this space, such as individual demonstrations in EPR or sampled demonstration sets in CEIL (Rubin et al., 2022; Ye et al., 2023). This feedback reflects the scoring model, motivating the transfer analysis below. For skill-augmented reasoning, we examine whether preferred reasoning plans transfer across models.

Mean accuracy drop under transfer. For a query q, let $\mathcal{B}_q$ contain the evaluated numbers of demonstrations in ICL or reasoning plans in skill-augmented reasoning. In ICL, a number of demonstrations specifies a prefix length under each model's recorded retrieval pipeline. We write $u_\theta(q, b)$ for the observed correctness of the response produced by model θ using prefix or plan b. Correctness is 1 for a correct answer and 0 otherwise. For skill-augmented reasoning, we average correctness over three runs. For source model s, the choices attaining the highest utility form $\mathcal{B}_s^*(q) = \arg\max_{b \in \mathcal{B}_q} u_s(q, b)$. The target model t's accuracy drop relative to its own best performance is

$$\bar{L}_{s \to t}(q) = \max_{b \in \mathcal{B}_q} u_t(q, b) - \frac{1}{|\mathcal{B}_s^*(q)|} \sum_{b \in \mathcal{B}_s^*(q)} u_t(q, b). \tag{23}$$

We weight source-model ties equally, average the drop within each task, and then weight tasks equally. This analysis uses evaluation labels.

For ICL, we match queries and evaluated numbers of demonstrations across all five models, excluding requests shortened by the context limit and using each model's recorded retrieval pipeline. Table 23 reports representative transfer directions with 95% paired query-bootstrap intervals from 10,000 resamples within each task.

Table 23: Mean accuracy drop when transferring numbers of demonstrations or reasoning plans across models, relative to the target model's best evaluated choice. Target accuracy (%) is shown using its own best choices → choices transferred from the source model. Drops and 95% paired query-bootstrap intervals are in percentage points.

Setting	Source → target	Target accuracy (%)	Drop [95% CI]
ICL	Llama 3 8B → Qwen2.5 7B	93.58 → 76.42	17.16 [16.26, 18.09]
ICL	Qwen2.5 7B → Llama 3 8B	92.01 → 69.98	22.03 [21.02, 23.08]
Skill	Qwen3-4B → Qwen3-8B	87.49 → 74.92	12.57 [11.31, 13.86]
Skill	Qwen3-8B → Qwen3-4B	86.76 → 71.30	15.46 [14.23, 16.73]

E.2 QUERY-ADAPTIVE CONSENSUS DECISIONS

We keep each query's model responses fixed and transfer a decision made for another query within the same task. For ICL, this decision comprises the AG ordering subset and position, and the CM ordering subset and position. Donor and recipient queries have the same evaluated numbers of demonstrations, so the transferred decision can be applied to the recipient's own sequences.

For skill-augmented reasoning, we transfer the index of the response returned after consensus and score the recipient's own response at that index. This tests whether the final response choice adapts to the query. The transferred index identifies a no-plan or plan-conditioned response, not AG and CM's internal subsets and positions.

Own uses the same six-response outputs and aggregation settings as Table 4, detailed in Appendix A.3. Each set contains the no-plan response and five plan-conditioned responses. Shuffling stays within each task, model, inference run, and plan-addition order. Both Own and Shuffled average over six cyclic plan-addition orders and three runs, with equal weights across the four domains.

We use 1,000 permutations in which each query receives a decision from a different query, excluding groups containing only one query.

Evaluation and uncertainty. Paired differences use 10,000 query-bootstrap replicates within task for ICL and 20,000 for skill. Repeated runs are averaged within each query. These intervals quantify query-sampling uncertainty conditional on the saved model responses and decision rules.

Table 24 reports the uncertainty in the Own–Shuffled differences from Table 6. All three intervals exclude zero. Differences are computed before rounding the displayed accuracies.

Table 24: Accuracy gains from using each query's own consensus decision rather than a shuffled decision. Differences and 95% paired query-bootstrap intervals are in percentage points.

Setting	Model	Own − Shuffled	95% CI
ICL	Llama 3 8B	1.32	[0.59, 2.11]
Skill	Qwen3-4B	5.62	[4.42, 6.90]
Skill	Qwen3-8B	7.75	[6.42, 9.13]

E.3 Effects of adding demonstrations across queries

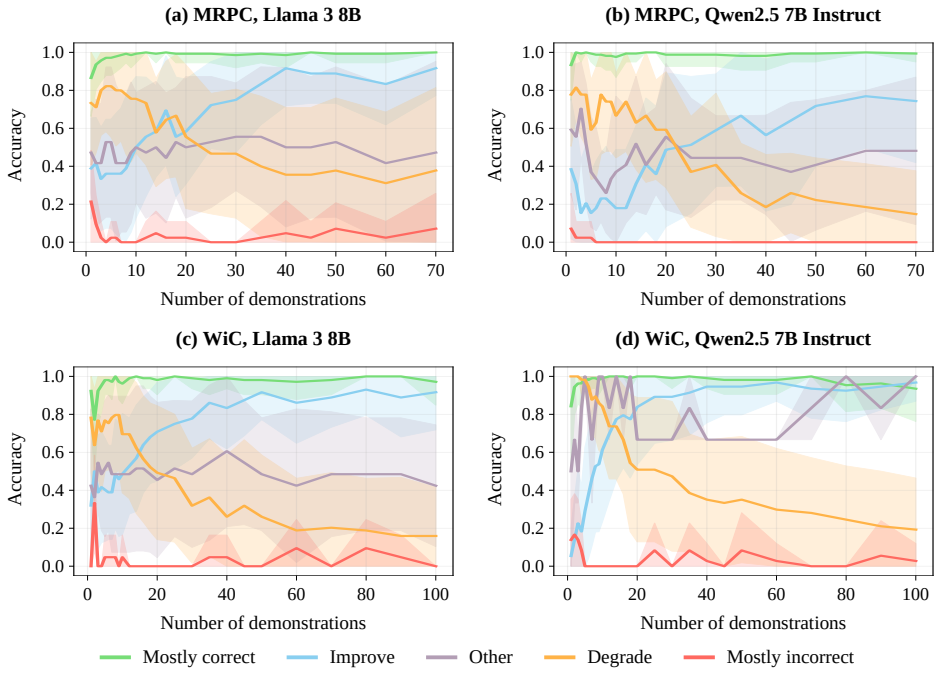


Figure 4: Mean accuracy within each query group. Rows show MRPC and WiC, and columns show Llama 3 8B and Qwen2.5 7B Instruct. Query accuracy is averaged over three orderings before grouping.

We examine MRPC and WiC as illustrative examples on Llama 3 8B and Qwen2.5 7B Instruct. For each query, we retrieve a candidate demonstration pool using BERT similarity (Devlin et al., 2019), vary the number of demonstrations k, and average correctness over three sampled orderings. Figure 4 summarizes how query accuracy changes as demonstrations are added.

We first assign queries with mean accuracy above 0.9 across evaluated positions to *Mostly correct*, and those below 0.1 to *Mostly incorrect*. For the remaining queries, we split the n evaluated positions into an initial half of $\lfloor n/2 \rfloor$ positions and the remaining positions. A rise of more than 0.15 in mean accuracy between the two halves defines *Improve*, and a fall of more than 0.15 defines *Degrade*. All remaining queries form the *Other* group, which includes patterns without a clear increase or decrease between the two halves. The figure illustrates that adding demonstrations can help some queries while hurting others within the same task and model. These descriptive groups use evaluation labels and do not enter CHORUS's decisions.

F COMPUTATIONAL COST

F.1 IN-CONTEXT LEARNING

The ICL methods compared here do not perform the same computational role. Demonstration selectors primarily construct a context before prediction, whereas CHORUS evaluates the target model under multiple context compositions and combines the resulting responses. The two are therefore not interchangeable methods under a matched inference budget. We report their costs together to characterize where computation is spent when improving the same downstream ICL prediction task. Selection methods may incur one-time retriever training or demonstration scoring and then use a selected context at inference, while CHORUS avoids a separate selector but performs additional target-model evaluations for each query.

Table 25 reports these cost profiles for Llama 3 8B on a single NVIDIA A6000 GPU, averaged equally over 12 tasks. The table should therefore be read as a comparison of preparation and query-time computation rather than as a head-to-head efficiency benchmark.

Table 25: Measured preparation and inference costs for the evaluated ICL approaches. The methods operate at different stages of the inference pipeline and are not compute-matched. Training/scoring is the one-time cost of retriever training or demonstration scoring in hours per task, and inference time is measured in seconds per query.

Method	EPR	EASE	Se2	SeDPO	DR-IKE	D.Va	CHORUS	CHORUS early stopping
Training/scoring (h)	2.27	0.22	8.07	10.85	0.10	–	–	–
Inference (s/query)	1.25	0.20	0.22	0.22	0.34	9.18	16.06	4.30
Accuracy (%, 12 tasks)	67.55	65.16	63.75	67.44	60.66	66.95	71.92	70.57

Optional early stopping ends an ordering's evaluation after the predicted class remains unchanged for 10 consecutive evaluated values of k, excluding unevaluated positions from consensus. This stopping rule is separate from AG and CM's subsequent evidence selection. It reduces inference time by 73%, from 16.06 to 4.30 seconds per query, with accuracy decreasing from 71.92% to 70.57%. Full CHORUS accuracy uses the main results' saved-response aggregation, while early-stopping accuracy is measured in the timed run.

F.2 SKILL-AUGMENTED REASONING

Unlike the ICL comparison, the skill-augmented reasoning methods use the same target model, evaluation queries, and supplied gold documents, allowing a direct comparison under matched response budgets. Figure 2 shows that CHORUS achieves higher mean accuracy than both Self-Consistency variants at the common fixed response budgets.

Table 26 asks a complementary question of how many responses CHORUS needs to reach the accuracy of each baseline. We report the smallest evaluated CHORUS budget that meets or exceeds the baseline accuracy, without interpolation, while Adaptive-Consistency uses its measured average number of responses under adaptive stopping. CHORUS reaches the corresponding baseline accuracy with fewer generated responses on both models. We interpret this result as response-generation efficiency rather than a hardware-normalized computational comparison, since response lengths and decoding procedures differ across methods.

G ADDITIONAL RELATED WORK

Demonstration composition. Beyond the methods discussed in Section 2, task-aware selection uses task or explanation information (An et al., 2023; Li et al., 2025b; Ghosal et al., 2025), while structure- and coverage-aware approaches seek complementary examples (Li et al., 2025a; Kothyari et al., 2025; Tang et al., 2024). Other work uses entropy or gradient information for selection (Wu et al., 2023; Peng et al., 2024; Zhang et al., 2025b), learns sequential selection policies (Wang et al., 2025; Scarlatos & Lan, 2023), or improves the ordering within a selected set (He et al., 2025; Li

Table 26: Response budgets required to reach baseline accuracy in skill-augmented reasoning. CHORUS uses the smallest evaluated budget that matches or exceeds each baseline's mean accuracy across four domains.

Baseline	Accuracy (%)		Responses per query		Fewer responses (%)
	Baseline	CHORUS	Baseline	CHORUS	
Qwen3-4B					
SC (weighted)	77.55	77.80	6.00	**4**	**33.3**
AC	77.33	77.80	4.57	**4**	**12.5**
Qwen3-8B					
SC (weighted)	76.43	77.08	6.00	**3**	**50.0**
AC	76.21	77.08	4.78	**3**	**37.3**

et al., 2024a). These approaches address how demonstrations are composed. Our study additionally examines how predictions under alternative compositions contribute to the final answer.

Understanding and extending context use. Representation analyses distinguish label separability from alignment (Yang et al., 2025b) and examine where task recognition emerges in the model (Sia et al., 2024). Analyses of domain-label bias and demonstration labels further show that supplying examples does not eliminate model or data biases (Fei et al., 2023; Kossen et al., 2024). These findings provide complementary perspectives on model-dependent context utility. Multimodal demonstrations (Zong et al., 2025; Li et al., 2024b) and demonstration compression (Li et al., 2024c; Huang et al., 2024) extend context use beyond our setting.