

Reason Analogically via Cross-domain Prior Knowledge: An Empirical Study of Cross-domain Knowledge Transfer for In-Context Learning

Anonymous ACL submission

Abstract

Despite its success, existing in-context learning (ICL) relies on in-domain expert demonstrations, limiting its applicability when expert annotations are scarce. We posit that different domains may share underlying reasoning structures, enabling source-domain demonstrations to improve target-domain inference despite semantic mismatch. To test this hypothesis, we conduct a comprehensive empirical study of different retrieval methods to validate the feasibility of achieving cross-domain knowledge transfer under the in-context learning setting. Our results demonstrate conditional positive transfer in cross-domain ICL. We identify a clear example absorption threshold: beyond it, positive transfer becomes more likely, and additional demonstrations yield larger gains. Further analysis suggests that these gains stem from reasoning structure repair by retrieved cross-domain examples, rather than semantic cues. Overall, our study validates the feasibility of leveraging cross-domain knowledge transfer to improve cross-domain ICL performance, motivating the community to explore designing more effective retrieval approaches for this novel direction.¹

1 Introduction

In-context learning (ICL) (Brown et al., 2020; Radford et al., 2019) allows LLMs to adapt to new tasks using only a small set of demonstrations, without any parameter fine-tuning (Dong et al., 2024). This flexibility has made ICL a central paradigm, motivating extensive in-domain studies on prompt design, example selection, and robustness to mild distribution shifts (Mueller et al., 2023; Zhou et al., 2023; Wei et al., 2022; Lewkowycz et al., 2022; Tang et al., 2023; Sun et al., 2024; Siska et al., 2024; Yuan et al., 2024; Honda and Oka, 2025; He et al., 2024). However, most existing studies

assume access to high-quality in-domain demonstrations curated by human experts. In practice, such annotations are often scarce or unavailable; a natural idea would be to leverage available prior knowledge which is embedded in other domains to improve performance.

As is often the case, different domains can share similar task-solving rationale despite substantial surface-level differences (Besta et al., 2024, 2025; Bu et al., 2025; Zhang et al., 2024; Li et al., 2024). Consequently, a well-designed retrieval method is indispensable for enabling effective transfer learning for ICL. To achieve this goal, retrieval methods must address two major outstanding challenges:

- **Retrieval Expressiveness:** Is the retrieval method sufficiently expressive to retrieve cross-domain demonstrations that share similar task-solving rationales?
- **Transfer Stability:** Does the retrieval method remain consistently effective across different large language models (LLMs)?

To demonstrate the feasibility of cross-domain transfer learning for LLMs and explore its future potential, we conduct a large-scale empirical study of existing, widely used retrieval methods for cross-domain in-context learning under a unified evaluation framework. Specifically, top-k demonstrations retrieved from source tasks are used to prompt a frozen LLM for target queries. Across six reasoning benchmarks and multiple model families and scales, we evaluate performance using Exact Match and gains over zero-shot. Our study is structured around four core research questions: ❶ How is the overall performance of cross-domain knowledge transfer via ICL? ❷ How does model scale influence cross-domain transferability?—We analyze how smaller and larger models differ in their ability to leverage retrieved demonstrations, including susceptibility to negative transfer. ❸ How does the

¹Our implementation is available at <https://anonymous.4open.science/r/Logical-Reasoning-related-E712>

number of shots (ie, demonstrations) affect cross-domain transferability?—We examine whether increasing shot counts consistently improves performance or instead introduces instability and interference. ④ What’s the source of improvement?—We unveil that the source of improvement stems from models’ rectification of reasoning structure based on cross-domain demonstrations.

The contributions of this work are summarized as follows.

- We conduct a systematic empirical study that examines the effectiveness of different retrieval approaches, their applicability across model families, and their transferability across domains in the setting of cross-domain in-context learning.
- We conduct detailed ablation analysis regarding the number of demonstrations and model size, which provides an in-depth understanding of the influence of different variables involved.
- To the best of our knowledge, this is the first trial to explore the feasibility of cross-domain transfer learning via in-context learning. These comprehensive evaluations may motivate the community to explore more effective retrieval methods that explicitly model logical structures and better support cross-domain generalization.

2 Preliminaries

2.1 Transfer Learning

Transfer learning studies how knowledge from a source dataset can be leveraged to improve performance on a target dataset, particularly when labeled target data are scarce. Let $\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$ and $\mathcal{D}_T = \{(x_j^T, y_j^T)\}_{j=1}^{n_T}$ denote the source and target datasets, where $(x, y) \in \mathcal{X} \times \mathcal{Y}$. In general, the source and target data may follow different input distributions,

$$P_S(x) \neq P_T(x),$$

and may correspond to different prediction functions.

The goal of transfer learning is to exploit information learned from \mathcal{D}_S to improve generalization on \mathcal{D}_T . A common approach is to map both source and target inputs into a shared representation space

via an encoder $h_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$, such that transferable structures are preserved:

$$\min_{\theta} \mathcal{L}_S(f_\theta(h_\theta(x^S)), y^S) + \lambda \mathcal{R}(\{h_\theta(x_i^S)\}, \{h_\theta(x_j^T)\}). \quad (1)$$

where \mathcal{L}_S is the source supervision loss and \mathcal{R} encourages cross-dataset transferability.

2.2 Retrieval-Augmented Generation

Retrieval-Augmented Generation enhances model inference by retrieving relevant documents from a corpus $\mathcal{D} = d_{i=1}^N$ via vector similarity:

$$\phi : \mathcal{D} \rightarrow \mathbb{R}^k,$$

and indexed into a retrieval database $\mathcal{I} = \{\phi(d_i)\}_{i=1}^N$. The top- K documents are retrieved based on the query representation $\phi(q)$:

$$\mathcal{R}(q) = \arg \text{topK}_{d_i \in \mathcal{D}} \text{sim}(\phi(q), \phi(d_i)),$$

where $\text{sim}(\cdot, \cdot)$ denotes a similarity measure such as inner product or cosine similarity.

The retrieved documents $\{d_{(1)}, \dots, d_{(K)}\}$ are combined with the query via a direct sum operation to form the augmented input:

$$\tilde{q} = d_{(1)} \oplus d_{(2)} \oplus \dots \oplus d_{(K)} \oplus q,$$

which is then fed into the language model to produce the final prediction:

$$y = f_\theta(\tilde{q}).$$

In this framework, retrieval serves as an explicit information access mechanism, enabling the model to condition its reasoning on external evidence rather than relying solely on parametric knowledge.

2.3 In-Context Learning

In-context learning (ICL) allows LLMs to infer new tasks from contextual examples (Wei et al., 2022; Brown et al., 2020). Unlike in-weights learning, which relies on gradient-based parameter updates, ICL adapts behavior without modifying model weights.

Formally, each training instance is linearized into an input sequence $\mathbf{x} = (x_1, \dots, x_{|\mathbf{x}|})$ and an output sequence $\mathbf{y} = (y_1, \dots, y_{|\mathbf{y}|})$, where each token belongs to the model vocabulary \mathcal{V} . Given

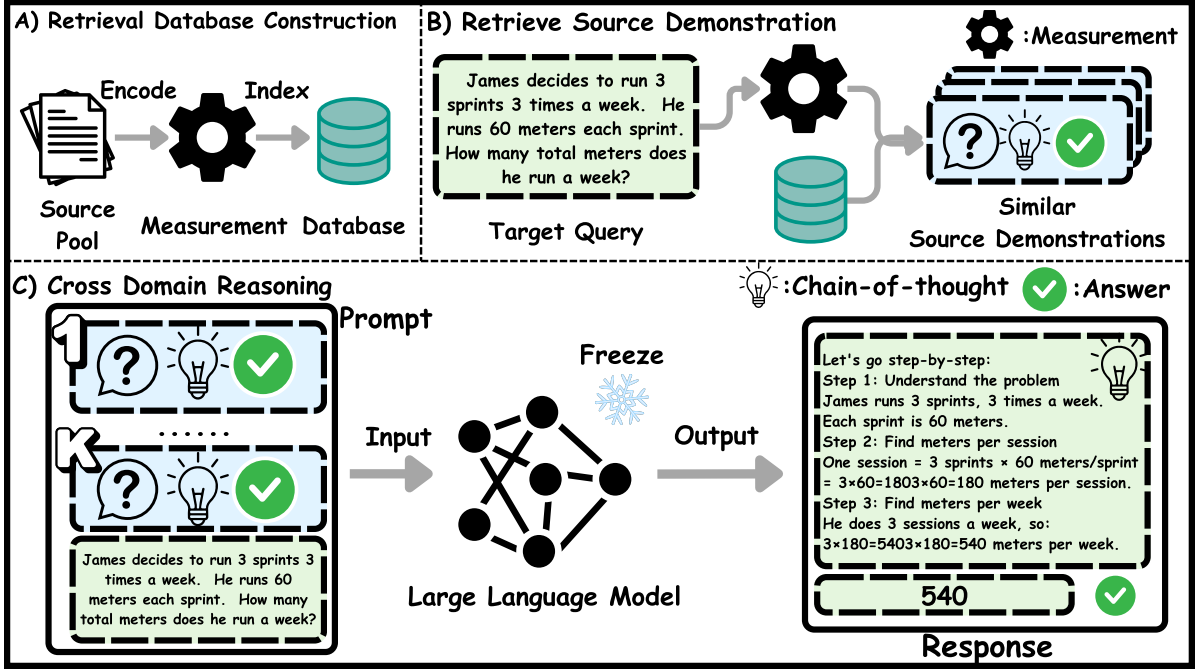


Figure 1: **Overview of our cross-domain ICL evaluation workflow.** (A) A source demonstration pool is encoded and indexed into a retrieval database. (B) For a target query, the system retrieves semantically similar source demonstrations. (C) Retrieved demonstrations are composed into a prompt and fed to a frozen LLM to produce step-by-step reasoning and the final answer.

a test input \mathbf{x}_{test} , in-context learning defines its prediction as

$$y_{\text{test}} \sim \mathcal{P}_{\text{LM}} \left(y_{\text{test}} \mid \underbrace{\mathbf{x}_1, \mathbf{y}_1, \dots, \mathbf{x}_K, \mathbf{y}_K, \mathbf{x}_{\text{test}}}_{\text{In-context prompt}} \right),$$

where the sampling operator denotes the decoding method. Each demonstration $e_i = (\mathbf{x}_i, \mathbf{y}_i)$ is drawn from a dataset

$$D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N.$$

This formulation allows the model to condition on the provided examples without updating its parameters, enabling fast adaptation to new tasks without additional training cost.

3 Method

As shown in Figure 1, we investigate the feasibility of cross-domain knowledge transfer using in-context learning through a three-step pipeline: embedding database construction, demonstration retrieval, and cross-domain inference.

3.1 Retrieval Database Construction

As shown in Part A of Figure 1, given a **source-domain dataset** $\mathcal{D}_S = \{(x_i^S, y_i^S)\}$, we construct a retrieval metric which embeds each instance to a

corresponding representation for the downstream retrieval function to each instance:

$$m_i = \mathcal{M}(x_i^S, y_i^S),$$

where $\mathcal{M}(\cdot)$ denotes a retrieval metric function that maps each source example into a searchable form. The resulting database is defined as

$$\mathcal{B}_S = \{(m_i, x_i^S, y_i^S)\},$$

which supports efficient retrieval under different similarity or matching criteria.

3.2 Demonstration Retrieval

As shown in Part B of Figure 1, given a target query x_{test}^T , we first compute its corresponding representation

$$m_{\text{test}}^T = \mathcal{M}(x_{\text{test}}^T),$$

where $\mathcal{M}(\cdot)$ denotes the same measurement function used in database construction.

We retrieve source demonstrations by ranking all indexed instances according to a similarity function $s(m_{\text{test}}^T, m_i)$ and selecting the top- k results:

$$\mathcal{R}(x_{\text{test}}^T) = \text{TopK}_{s(\cdot, \cdot)}(m_{\text{test}}^T, \mathcal{B}_S).$$

The similarity function $s(\cdot, \cdot)$ is used to measure the semantic distance between query and documents; inner-product, cosine similarity, and BM25 are all viable candidates.

Model	Source	Method	ProntoQA	FOLIO	ProofWriter	Logical Deduction	AR-LSAT	GSM8K	
Gemma3-4B	ProntoQA	BM25	—	60.8 _{-2.5}	66.2 _{+2.7}	36.7 _{-19.7}	—	—	
		Embed	—	66.2 _{+2.9}	67.8 _{+4.3}	58.3 _{+2.0}	25.7 _{-2.6}	84.8 _{-1.3}	
		ConE	—	60.8 _{-2.5}	62.5 _{-1.0}	63.3 _{+7.0}	26.1 _{-2.2}	85.2 _{-0.9}	
	FOLIO	BM25	93.2 _{+1.2}	—	63.0 _{+1.3}	41.3 _{-19.7}	24.8 _{-0.4}	84.0 _{-2.4}	
		Embed	93.0 _{+1.0}	—	65.0 _{+3.3}	58.7 _{-2.3}	27.0 _{+1.7}	85.0 _{-1.4}	
		ConE	94.4 _{+2.4}	—	64.3 _{+2.7}	62.7 _{+1.7}	30.0 _{+4.8}	—	
	ProofWriter	BM25	95.2 _{+2.4}	67.6 _{+5.4}	—	41.3 _{-19.3}	28.7 _{+3.5}	85.7 _{-0.4}	
		Embed	95.6 _{+2.8}	65.2 _{+2.9}	—	61.0 _{+0.3}	26.1 _{+0.9}	84.5 _{-1.6}	
		ConE	—	65.2 _{+2.9}	—	62.3 _{+1.7}	27.0 _{+1.7}	—	
	LogicalDeduction	BM25	65.4 _{-27.6}	63.7 _{+2.0}	60.8 _{-1.8}	—	27.0 _{-0.9}	84.1 _{-1.8}	
		Embed	90.6 _{-2.4}	63.7 _{+2.0}	61.5 _{-1.2}	—	26.1 _{-1.7}	84.5 _{-1.4}	
		ConE	89.2 _{-3.8}	61.8	56.2 _{-6.5}	—	28.3 _{+0.4}	—	
	GSM8K	BM25	86.6 _{-6.0}	62.7 _{+1.0}	60.5 _{-3.8}	62.7	26.1 _{-2.6}	—	
		Embed	86.6 _{-6.0}	60.8 _{-1.0}	58.2 _{-6.2}	63.0 _{+0.3}	27.4 _{-1.3}	—	
		ConE	86.8 _{-5.8}	62.3 _{+0.5}	57.8 _{-6.5}	64.7 _{+2.0}	26.5 _{-2.2}	—	
	Gemma3-12B	ProntoQA	BM25	—	71.6 _{-1.5}	64.8 _{-9.0}	64.0 _{-9.0}	31.7 _{-1.3}	91.3 _{-0.8}
			Embed	—	76.5 _{+3.4}	76.2 _{+2.3}	78.7 _{+5.7}	30.9 _{-2.2}	91.2 _{-0.8}
			ConE	—	71.6 _{-1.5}	65.2 _{-8.7}	75.7 _{+2.7}	36.1 _{+3.0}	91.6 _{-0.5}
FOLIO		BM25	99.4 _{+1.2}	—	77.5 _{+3.7}	66.7 _{-6.7}	35.2 _{+5.2}	91.1 _{-0.9}	
		Embed	98.8 _{+0.6}	—	76.5 _{+2.7}	78.0 _{+4.7}	33.5 _{+3.5}	90.8 _{-1.2}	
		ConE	98.8 _{+0.6}	—	77.8 _{+4.0}	77.0 _{+3.7}	33.0 _{+3.0}	—	
ProofWriter		BM25	99.6 _{+1.4}	78.9 _{+6.9}	—	67.0 _{-6.0}	32.6 _{-0.4}	91.0 _{-1.1}	
		Embed	99.2 _{+1.0}	77.5 _{+5.4}	—	78.3 _{+5.3}	32.6 _{-0.4}	91.0 _{-1.1}	
		ConE	—	77.9 _{+5.9}	—	80.0 _{+7.0}	33.5 _{+0.4}	—	
LogicalDeduction		BM25	94.2 _{-4.0}	75.0 _{+3.9}	75.7 _{+1.8}	—	32.6 _{+2.6}	92.0 _{+0.2}	
		Embed	98.0 _{-0.2}	73.0 _{+2.0}	76.2 _{+2.3}	—	30.9 _{+0.9}	91.7 _{-0.1}	
		ConE	99.0 _{+0.8}	74.0 _{+2.9}	76.8 _{+3.0}	—	32.2 _{+2.2}	—	
GSM8K		BM25	98.8 _{+0.6}	72.1	75.3 _{+1.5}	77.7 _{+4.7}	33.9 _{+0.9}	—	
		Embed	98.0 _{-0.2}	70.1 _{-2.0}	74.3 _{+0.5}	75.0 _{+2.0}	32.2 _{-0.9}	—	
		ConE	99.6 _{+1.4}	70.1 _{-2.0}	74.7 _{+0.8}	79.3 _{+6.3}	33.5 _{+0.4}	—	
Gemma3-27B		ProntoQA	BM25	—	75.0 _{+3.9}	77.3	84.0 _{+6.3}	38.7 _{-1.3}	93.1 _{-0.1}
			Embed	—	77.0 _{+5.9}	82.5 _{+5.2}	88.0 _{+10.3}	39.1 _{-0.9}	93.3 _{+0.1}
			ConE	—	73.5 _{+2.5}	75.7 _{-1.7}	87.0 _{+9.3}	37.8 _{-2.2}	93.4 _{+0.2}
	FOLIO	BM25	99.2 _{+0.8}	—	81.3 _{+4.0}	86.7 _{+9.0}	39.1 _{+4.3}	93.7 _{+0.5}	
		Embed	99.0 _{+0.6}	—	83.2 _{+5.8}	86.0 _{+8.3}	37.0 _{+2.2}	93.4 _{+0.2}	
		ConE	99.6 _{+1.2}	—	82.0 _{+4.7}	89.0 _{+11.3}	40.9 _{+6.1}	—	
	ProofWriter	BM25	100.0 _{+1.8}	76.5 _{+5.4}	—	87.0 _{+9.3}	40.4 _{+0.9}	93.6 _{+0.4}	
		Embed	99.6 _{+1.4}	78.4 _{+7.4}	—	90.0 _{+12.3}	40.0 _{+0.4}	93.6 _{+0.4}	
		ConE	—	78.9 _{+7.8}	—	87.7 _{+10.0}	40.0 _{+0.4}	—	
	LogicalDeduction	BM25	99.2 _{+0.8}	74.5 _{+3.4}	78.5 _{+2.5}	—	41.7 _{+1.7}	93.4 _{+0.2}	
		Embed	99.2 _{+0.8}	73.5 _{+2.5}	78.8 _{+2.8}	—	43.9 _{+3.9}	93.2 _{-0.1}	
		ConE	99.2 _{+0.8}	76.0 _{+4.9}	79.2 _{+3.2}	—	40.0	—	
	GSM8K	BM25	98.2	72.5 _{+1.5}	80.0 _{+2.7}	85.7 _{+7.0}	38.7 _{-0.9}	—	
		Embed	97.4 _{-0.8}	73.5 _{+2.5}	76.8 _{-0.5}	87.0 _{+8.3}	38.7 _{-0.9}	—	
		ConE	98.2	71.6 _{+0.5}	74.2 _{-3.2}	90.3 _{+11.7}	38.3 _{-1.3}	—	

Table 1: Relative performance of different retrieval strategies across source–target domain pairs, measured against the 0-shot baseline. Deeper blue indicates larger gains, while deeper red denotes greater degradation.

3.3 Cross-domain ICL

Finally, the retrieved source demonstrations $\{(x_i^S, y_i^S)\}_{i=1}^k$ and the target query x_{test}^T are concatenated to form the input prompt, the model prediction follows the standard ICL formulation (Part C in Figure 1):

$$\hat{y} \sim \mathcal{P}_{\text{LM}} \left(\underbrace{[(x_1^S, y_1^S), \dots, (x_k^S, y_k^S)]}_{\text{Source Domain}}, \underbrace{x_{\text{test}}^T}_{\text{Target Domain}} \right) \quad (2)$$

We evaluate accuracy on \mathcal{D}_T under varying

source domains, target domains, shot counts, and model sizes. This framework enables controlled and systematic analysis of factors that influence cross-domain ICL performance.

4 Experimental Setup

4.1 Models and Datasets

For models, we evaluate a diverse set of backbone models spanning multiple families and parameter scales, including **Qwen2.5** (3B, 7B, 14B, 32B) (Team, 2024), **Qwen3** (4B, 8B, 14B, 32B) (Team, 2025b), **Gemma 3** (4B, 12B, 27B) (Team, 2025a), and **Llama 3.1** (8B) (Grattafiori et al., 2024).

For datasets, we use six reasoning benchmarks: **GSM8K** (grade-school math) (Cobbe et al., 2021), **ProntoQA** (synthetic multi-hop deduction) (Saparov and He, 2022), **LogicalDeduction** (symbolic consistency reasoning) (Nguyen et al., 2025), **FOLIO** (first-order logical inference) (Han et al., 2022), **ProofWriter** (deductive proofs) (Tafjord et al., 2021), and **AR-LSAT** (law school analytical reasoning) (Zhong et al., 2021). These datasets span arithmetic, symbolic logic, rule-based deduction, and formal reasoning, providing a broad testbed for evaluating ICL transferability.

4.2 Baselines

We evaluate three representative retrieval metric functions for constructing in-context demonstrations: (1) **Embedding-based retrieval** (Lewis et al., 2021), which selects top-k examples via cosine similarity in the model’s embedding space (We choose bge-large-en-v1.5 as the embedding model (Xiao et al., 2023)); (2) **BM25 retrieval** (Robertson and Zaragoza, 2009), a lexical matching baseline widely used in IR and prior ICL studies; (3) **TopK+ConE** (Peng et al., 2024) which refines retrieved candidates through conditional-entropy-based reranking to identify the most informative demonstrations.

5 Main Results

5.1 RQ1: How is the overall performance of cross-domain knowledge transfer via ICL?

In order to investigate the overall performance of cross-domain knowledge transfer via ICL, we conducted large-scale experiments using three representative baselines across 25 different transfer directions. Table 1 reveals a clear scaling trend: as model size increases, cross-domain knowledge

transfer becomes consistently more beneficial, with negative transfer substantially reduced. In particular, Gemma3-27B exhibits negative transfer in only five source–target pairs, whereas smaller variants (4B and 12B) suffer from considerably more failures.

The gains are especially pronounced on the Logical Deduction benchmark. For Gemma3-27B, embedding-based retrieval improves accuracy by +12.3% for ProofWriter→Logical Deduction and by +10.3% for ProntoQA→Logical Deduction.

We attribute this effect to model scaling: smaller models often fail to abstract task-solving logic from cross-domain demonstrations and are distracted by surface-level mismatches, leading to negative transfer. In contrast, larger models can better identify and exploit shared reasoning structures across domains. Figure 2 illustrates a representative Logical Deduction example where Gemma3-27B recovers the correct reasoning chain after conditioning on an embedding-retrieved cross-domain demonstration.

Q I: Cross-domain ICL shows systematic and often substantial positive transfer in specific source–target directions.

5.2 RQ2: How does model scale influence cross-domain transferability?

Figure 3 reveals a clear scaling-dependent pattern in cross-domain ICL. Small models (3B–7B) exhibit highly unstable behaviour, where negative transfer is common, and the average Spearman correlation between shot size and performance is near zero or even negative. This indicates that additional demonstrations are often not “absorbed” effectively, and may even introduce structural mismatches that degrade reasoning.

As model size increases, the trend shifts: mid- to large-scale models (12B–32B) show consistently positive correlations and markedly fewer negative-transfer cases. These models not only benefit more reliably from additional examples but also maintain stable improvements across a broader range of source–target directions.

Taken together, the results suggest the presence of an *example absorption threshold*: only models above a certain capacity can reliably leverage cross-domain demonstrations, while smaller models remain highly sensitive and prone to degradation. This highlights model scale as a key factor governing whether ICL exhibits positive or negative shot

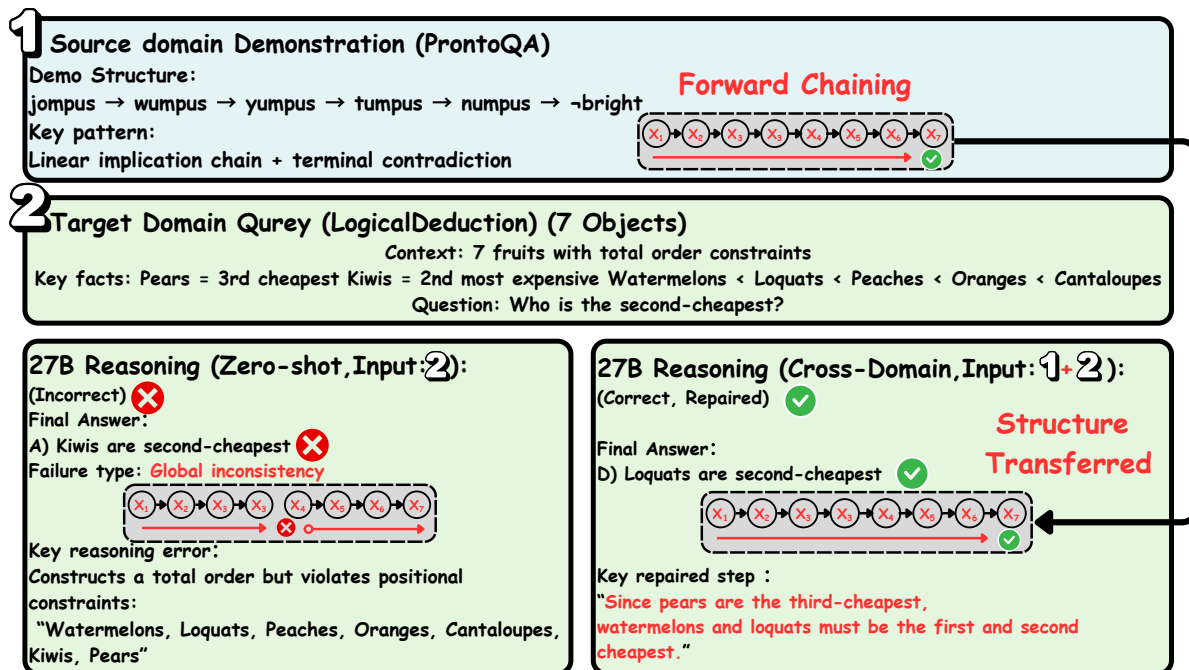


Figure 2: Comparison between zero-shot and cross-domain ICL. Bottom left: Zero-shot reasoning omits a required intermediate link, leading to an incorrect prediction. Bottom right: Cross-domain ICL restores the missing link via a structurally compatible demonstration, yielding the correct answer.

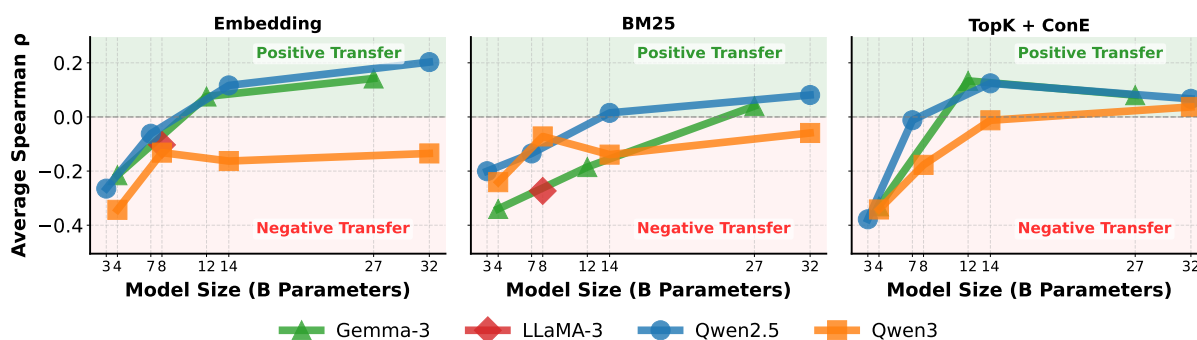


Figure 3: Scaling behaviour of cross-domain transfer under different retrieval baselines. Average Spearman ρ between model size and few-shot performance is plotted for Embedding and BM25 across model families. Shaded regions denote positive and negative transfer.

scaling.

❓ II: Cross-domain ICL exhibits an absorption threshold: sufficiently large models consistently benefit from cross-domain demonstrations, while smaller models are prone to negative transfer.

5.3 RQ3: How does the number of shots affect cross-domain transferability?

Figure 4 further examines how model capacity mediates the effect of demonstration quantity in cross-domain ICL. We observe a clear divergence between large and small models in their ability to benefit from additional in-context examples.

For the 27B model, performance consistently improves with more demonstrations across all positive transfer pairs, indicating strong capacity for cross-domain integration. In contrast, smaller models (4B and 12B) show unstable scaling behavior, where additional demonstrations often lead to diminishing or even negative returns, especially on structurally mismatched targets such as GSM8K and AR-LSAT. These results suggest that while larger models can effectively exploit richer cross-domain supervision, smaller models lack the representational capacity to accommodate numerous heterogeneous examples, leading to interference rather than benefit.

Overall, the findings indicate that the effective-

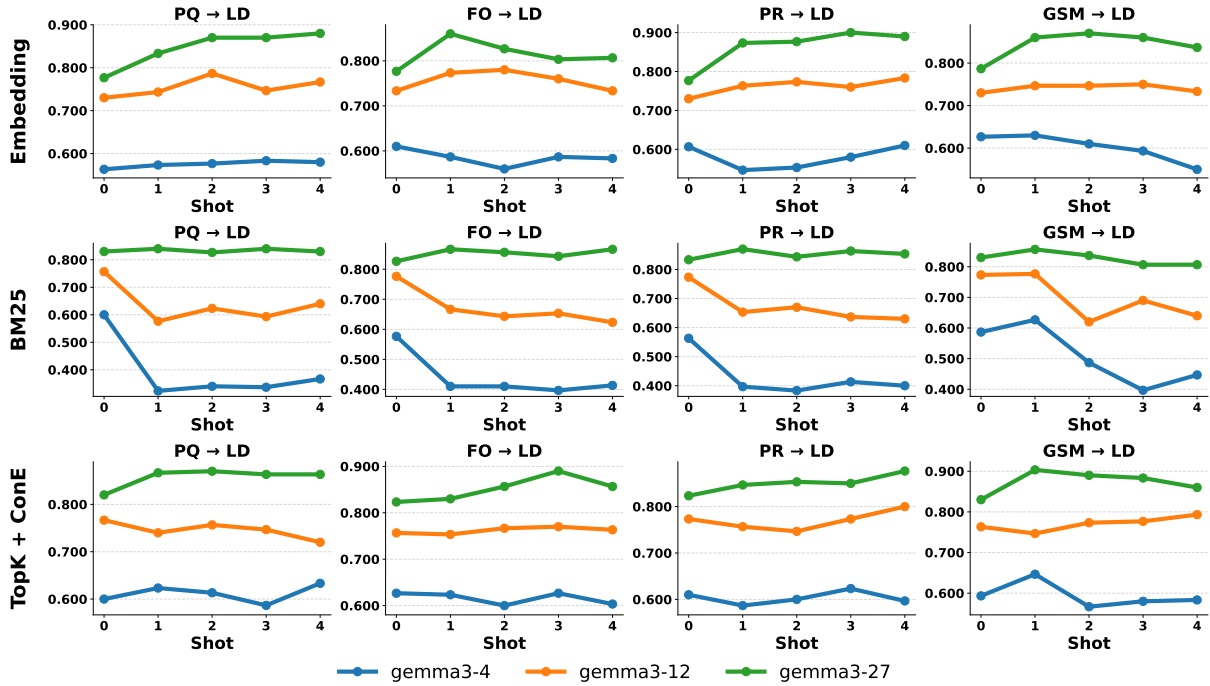


Figure 4: Shot-performance scaling across cross-domain ICL settings.

ness of increasing shot size is strongly dependent on model scale. While larger models can better leverage additional demonstrations, smaller models often exhibit diminishing or negative returns, indicating that simply increasing the number of demonstrations does not reliably improve performance.

III: Once the source-target transfer exceeds an absorption threshold, increasing the shot size within a moderate range yields consistently larger cross-domain ICL gains.

5.4 RQ4: What’s the source of improvement?

To better understand the source of improvement, we further conduct an in-depth analysis based on the categorization of task-solving logic. Concretely, we first collect the samples that the Gemma3-27b model originally did wrong, and deduce correctly with the introduction of cross-domain demonstrations. As shown in Figure 5, we categorize this set of samples into four categories according to the structures of their task-solving logic, namely L-type (A), Y-type (B), Block-type (C), and other types (D). We then used Deepseek-V3 (DeepSeek-AI, 2024) to classify the inference structure of the target query and the retrieved source domain examples. Finally, we plotted the logic structure distribution of all retrieval examples that repaired 0-shot samples when LogicalDeduction was the target do-

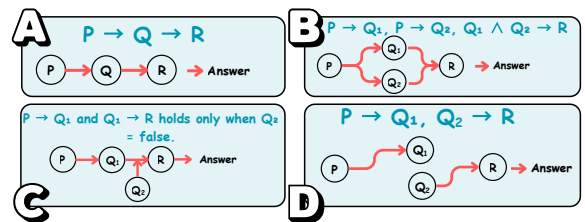


Figure 5: Four types of forward chaining.

main.

As shown in Figure 6, the repair ratio varies considerably across retrieval methods and transfer directions. For instance, under embedding-based retrieval, Query Type A achieves repair ratios of 52.6%–65.8%, whereas the same query type under BM25 drops sharply to 18.4%–26.3%. Similar sensitivity is observed for Query Type D, where repair ratios remain below 36.8% and fall to as low as 5.3% under mismatched retrieval-demo configurations. These results indicate that cross-domain ICL gains are highly sensitive to the retrieval configuration and depend critically on whether the retrieved demonstrations align with the underlying reasoning structure of the query.

At the demonstration level, the retrieved demonstrations for repaired queries exhibit a clear topology bias. Both embedding-based retrieval and BM25 tend to return demonstrations dominated by a small number of chain types (e.g., type A), while other types appear rarely. Such biased retrieval suggests that current retrievers do not fully

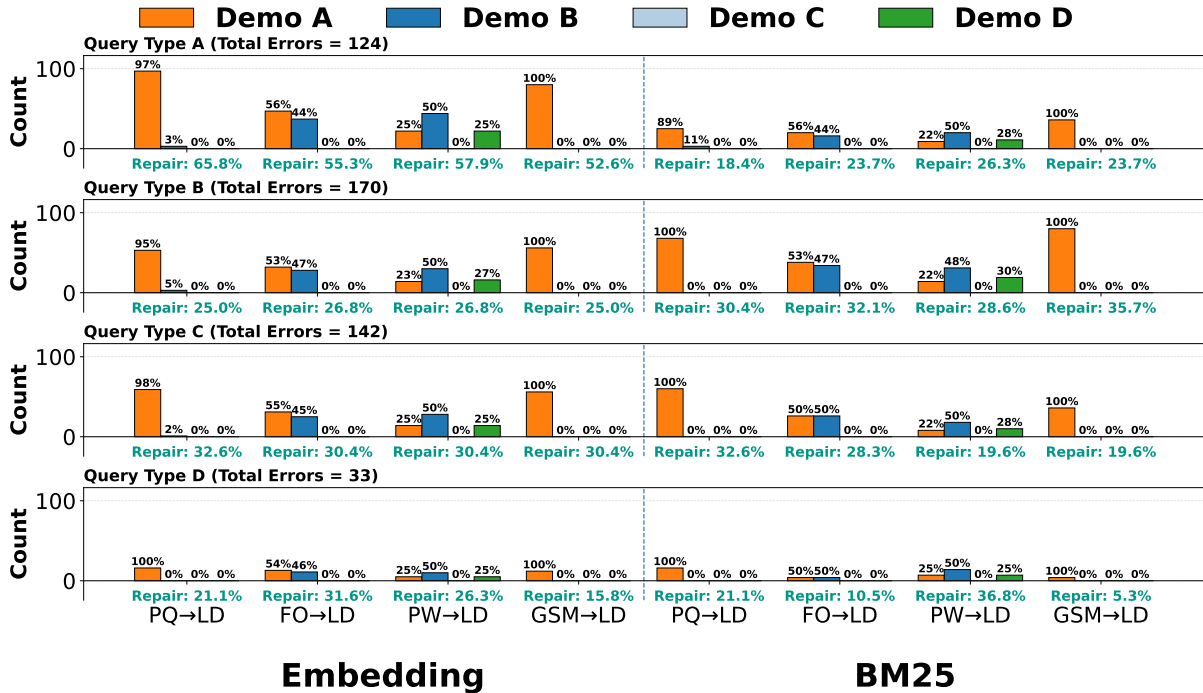


Figure 6: Distribution of demonstration topologies among repaired zero-shot errors across retrieval methods and transfer directions.

explore structurally diverse yet potentially transferable demonstrations, which may limit the achievable cross-domain transfer gains. This points to a promising future direction: incorporating chain-structure similarity into retrieval to better align demonstrations with target reasoning structures and improve cross-domain performance.

🔍 IV: Cross-domain ICL gains arise from retrieval-induced repair of a subset of zero-shot failures via structurally compatible demonstrations.

6 Related Work

6.1 Generalization in LLM

Large language models (LLMs) often degrade under domain shifts (Öncel et al., 2024; Oh et al., 2025). Existing approaches—such as data-centric adaptation (Wang et al., 2024), prompt calibration (Zhao et al., 2021; Honda and Oka, 2025; He et al., 2024), and parameter-efficient tuning (Hu et al., 2022). Recent work has begun examining cross-domain representation alignment (Aghajanyan et al., 2020), including neuron-level alignment in multilingual settings (Huang et al., 2025). Neuron-level analyses are well studied (Chen et al., 2024; Sajjad et al., 2022), but their role in cross-domain in-context learning remains unexplored.

6.2 Example Selection for In-Context Learning

In-context learning (ICL) is highly sensitive to demonstration selection (Luo et al., 2024). Existing retrieval methods rely on semantic similarity (Rubin et al., 2021), dense retrievers (Wang et al., 2023), uncertainty signals (Ling et al., 2024; Huang et al., 2024; Margatina et al., 2023), coverage-based selection (Gupta et al., 2023), or MMR-based diversification (Liu et al., 2023). Recent work uses internal representations to analyze LLM reasoning (Liu et al., 2023). Overall, prior work relies on heuristic retrieval and overlooks reasoning-level alignment.

7 Conclusion

We conducted a systematic empirical study of cross-domain in-context learning, showing that transfer performance is sensitive to retrieval quality and demonstration selection, and that only models above an example absorption threshold can reliably benefit from cross-domain demonstrations, while increasing the number of demonstrations does not consistently improve results. Our findings suggest that effective cross-domain transfer mainly stems from structurally compatible demonstrations that repair zero-shot failures, highlighting the need for structure-aware retrieval in future work.

432 Limitations

433 Our study is limited to a specific set of reason-
434 ing tasks and retrieval methods, and the findings
435 may not fully generalize to other domains or re-
436 trieval paradigms. The notion of reasoning repair
437 is characterized heuristically rather than through a
438 formally defined structural metric. In addition, our
439 analysis focuses on a fixed decoding setup, leav-
440 ing the interaction between retrieval effects and
441 different decoding strategies underexplored. Fi-
442 nally, while we observe complementary behaviours
443 across retrieval methods, we do not explicitly
444 model or optimize this complementarity.

445 References

446 Armen Aghajanyan, Luke Zettlemoyer, and Sonal
447 Gupta. 2020. Intrinsic dimensionality explains the
448 effectiveness of language model fine-tuning. *arXiv*
449 *preprint arXiv:2012.13255*.

450 Maciej Besta, Nils Blach, Ales Kubicek, Robert Ger-
451 stenberger, Michal Podstawski, Lukas Gianinazzi,
452 Joanna Gajda, Tomasz Lehmann, Hubert Niewiadoski,
453 Piotr Nyczyk, and Torsten Hoefler. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

454 Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert
455 Gerstenberger, Guangyuan Piao, Nils Blach, Piotr
456 Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadoski, Aidan O’Mahony, Onur Mutlu, and Torsten Hoefler. 2025. [Demystifying chains, trees, and graphs of thoughts](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):10967–10989.

466 Tom Brown, Benjamin Mann, Nick Ryder, Melanie
467 Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind
468 Neelakantan, Pranav Shyam, Girish Sastry, Amanda
469 Askell, Sandhini Agarwal, Ariel Herbert-Voss,
470 Gretchen Krueger, Tom Henighan, Rewon Child,
471 Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens
472 Winter, Chris Hesse, Mark Chen, Eric Sigler, Ma-
473 teusz Litwin, Scott Gray, Benjamin Chess, Jack
474 Clark, Christopher Berner, Sam McCandlish, Alec
475 Radford, Ilya Sutskever, and Dario Amodei. 2020.
476 [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

480 Tianpeng Bu, Mingyong Zhang, Hongtao Duan, Shurui
481 Li, Lulu Hu, and Yu Li. 2025. [Enhanced data synthesis for LLM through reasoning structures generated by hierarchical GFlowNet](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 15931–15958, Vienna, Austria. Association for Computational Linguistics.

Lihu Chen, Adam Dejl, and Francesca Toni. 2024. [Analyzing key neurons in large language models](#). *arXiv e-prints*, pages arXiv–2406. 487
488
489

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, 490
Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias 491
Plappert, Jerry Tworek, Jacob Hilton, Reiichiro 492
Nakano, Christopher Hesse, and John Schulman. 493
2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*. 494
495

DeepSeek-AI. 2024. [Deepseek-v3 technical report](#). 496
Preprint, arXiv:2412.19437. 497

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan 498
Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, 499
Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 500
2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics. 501
502
503
504
505

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, 506
and Abhinav Pandey. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783. 507
508

Shivanshu Gupta, Matt Gardner, and Sameer Singh. 509
2023. [Coverage-based example selection for in-context learning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics. 510
511
512
513
514

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting 515
Qi, Martin Riddell, Luke Benson, Lucy Sun, Eka- 516
terina Zubova, Yujie Qiao, Matthew Burtell, David 517
Peng, Jonathan Fan, Yixin Liu, Brian Wong, Mal- 518
colm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, 519
Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fab- 520
bri, Wojciech Kryscinski, Xi Victoria Lin, Caiming 521
Xiong, and Dragomir Radev. 2022. [Folio: Natural language reasoning with first-order logic](#). *arXiv preprint arXiv:2209.00840*. 522
523
524

Xuanli He, Yuxiang Wu, Oana-Maria Camburu, 525
Pasquale Minervini, and Pontus Stenetorp. 2024. [Using natural language explanations to improve robustness of in-context learning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13477–13499, Bangkok, Thailand. Association for Computational Linguistics. 526
527
528
529
530
531
532
533

Ukyo Honda and Tatsushi Oka. 2025. [Exploring explanations improves the robustness of in-context learning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23693–23714, Vienna, Austria. Association for Computational Linguistics. 534
535
536
537
538

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan 539
Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, 540
Weizhu Chen, et al. 2022. [Lora: Low-rank adaptation of large language models](#). *ICLR*, 1(2):3. 541
542

651	Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021.	Coates, Yingxue Zhang, and Jianye Hao. 2024. Path-	704
652	ProofWriter: Generating implications, proofs, and	of-thoughts: Extracting and following paths for ro-	705
653	abductive statements over natural language. In <i>Find-</i>	bust relational reasoning with large language models.	706
654	<i>ings of the Association for Computational Linguis-</i>	<i>Preprint</i> , arXiv:2412.17963.	707
655	<i>tics: ACL-IJCNLP 2021</i> , pages 3621–3634, Online.		
656	Association for Computational Linguistics.		
657	Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui	Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and	708
658	Xue. 2023. Large language models can be lazy learn-	Sameer Singh. 2021. Calibrate before use: Improv-	709
659	ers: Analyze shortcuts in in-context learning. <i>arXiv</i>	ing few-shot performance of language models. In	710
660	<i>preprint arXiv:2305.17256</i> .	<i>International conference on machine learning</i> , pages	711
661	Gemma Team. 2025a. Gemma 3 .	12697–12706. PMLR.	712
662	Qwen Team. 2024. Qwen2.5: A party of foundation	Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu,	713
663	models .	Daya Guo, Jiahai Wang, Jian Yin, Ming Zhou, and	714
664	Qwen Team. 2025b. Qwen3 technical report . <i>Preprint</i> ,	Nan Duan. 2021. Ar-lsat: Investigating analytical	715
665	arXiv:2505.09388.	reasoning of text . <i>Preprint</i> , arXiv:2104.06598.	716
666	Jinchao Wang, Qian Zhang, Jun Ma, and Xinyu Sun.	Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei	717
667	2024. Analysing neurons across languages and tasks	Ai, and Furong Huang. 2023. Explore spurious cor-	718
668	in large language models. In <i>Proceedings of the</i>	relations at the concept level in language models for	719
669	<i>2024 Conference on Empirical Methods in Natural</i>	text classification. <i>arXiv preprint arXiv:2311.08648</i> .	720
670	<i>Language Processing</i> .		
671	Liang Wang, Nan Yang, and Furu Wei. 2023. Learning		
672	to retrieve in-context examples for large language		
673	models. <i>arXiv preprint arXiv:2307.07164</i> .		
674	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten		
675	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,		
676	et al. 2022. Chain-of-thought prompting elicits reason-		
677	ing in large language models. <i>Advances in neural</i>		
678	<i>information processing systems</i> , 35:24824–24837.		
679	Thomas Wolf, Lysandre Debut, Victor Sanh, Julien		
680	Chaumond, Clement Delangue, Anthony Moi, Pier-		
681	ric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz,		
682	Joe Davison, Sam Shleifer, Patrick von Platen, Clara		
683	Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le		
684	Scao, Sylvain Gugger, Mariama Drame, Quentin		
685	Lhoest, and Alexander M. Rush. 2020. Transform-		
686	ers: State-of-the-art natural language processing . In		
687	<i>Proceedings of the 2020 Conference on Empirical</i>		
688	<i>Methods in Natural Language Processing: System</i>		
689	<i>Demonstrations</i> , pages 38–45, Online. Association		
690	for Computational Linguistics.		
691	Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas		
692	Muennighoff. 2023. C-pack: Packaged resources		
693	to advance general chinese embedding . <i>Preprint</i> ,		
694	arXiv:2309.07597.		
695	Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and		
696	Qi Liu. 2024. Do LLMs overcome shortcut learning?		
697	an evaluation of shortcut challenges in large language		
698	models . In <i>Proceedings of the 2024 Conference on</i>		
699	<i>Empirical Methods in Natural Language Processing</i> ,		
700	pages 12188–12200, Miami, Florida, USA. Associa-		
701	tion for Computational Linguistics.		
702	Ge Zhang, Mohammad Ali Alomrani, Hongjian Gu, Ji-		
703	aming Zhou, Yaochen Hu, Bin Wang, Qun Liu, Mark		

A Appendix

A.1 Prompt Construction for In-Context Learning

This section describes the prompt templates used for all in-context learning (ICL) experiments. Unless otherwise specified, the same prompt structure is applied across all models, tasks, and retrieval methods to ensure fair comparison.

A.1.1 Zero-Shot Prompt

For zero-shot inference, the prompt consists solely of a task instruction followed by the query instance:

$$\text{Prompt}_{0\text{-shot}} = [\text{Instruction}; x_q], \quad (3)$$

where x_q denotes the target query. The instruction provides a brief task description and remains fixed within each dataset.

A.1.2 k -Shot Prompt

For k -shot ICL, the prompt is constructed by concatenating k retrieved demonstration examples with the query:

$$\text{Prompt}_{k\text{-shot}} = [\text{Instruction}; d_{i_1}, d_{i_2}, \dots, d_{i_k}; x_q], \quad (4)$$

where $\{d_{i_j}\}_{j=1}^k$ are the top- k demonstrations returned by the retrieval module (Appendix A.4). Each demonstration d_i follows a fixed format:

$$d_i = [x_i; y_i], \quad (5)$$

with x_i denoting the input (e.g., context and question) and y_i the corresponding gold answer.

A.1.3 Demonstration Ordering

Retrieved demonstrations are ordered by decreasing retrieval score, such that d_{i_1} is the most similar example to the query. To study order sensitivity, we also evaluate a reversed order setting, where demonstrations are concatenated in ascending order of retrieval score. Apart from the ordering, all other prompt components are kept identical.

A.1.4 Reasoning and Answer Format

For tasks that involve explicit reasoning, demonstrations may include intermediate reasoning steps (e.g., chain-of-thought) followed by a final answer. The query instance, however, only requires the model to output the final answer. This design prevents information leakage while preserving the structural guidance provided by the demonstrations.

A.1.5 Prompt Consistency

All retrieval methods, shot numbers, and experimental conditions share the same prompt template and textual fields. The total prompt length is constrained by the model context window, and no additional prompt engineering or task-specific tuning is applied beyond the fixed templates described above.

A.2 Prompt Templates

We adopt task-specific *system prompts* for each dataset to ensure (i) a consistent reasoning style across domains and (ii) a unified, strictly formatted final answer for reliable automatic evaluation. All prompts follow a two-stage structure: the model is first instructed to provide a short step-by-step reasoning, and then output the final answer on a separate line using a standardized format (e.g., Final answer: A). This design minimizes ambiguity in option extraction and avoids dataset-specific parsing heuristics.

ProntoQA. ProntoQA is a binary logical reasoning task with two answer options. We use the following system prompt:

```
You are a careful reasoner. Think step by step concisely.
Then on a new line, output exactly: 'Final answer: A' or 'Final answer: B'.
```

FOLIO. FOLIO requires three-way classification (*entailment / contradiction / unknown*). We extend the same reasoning template to support three options:

```
You are a careful reasoner. Think step by step concisely.
Then on a new line, output exactly:
'Final answer: A' or 'Final answer: B' or 'Final answer: C'.
```

ProofWriter. ProofWriter is a logical reasoning task typically formulated as entailment-style verification. We adopt the same binary template as ProntoQA:

```
You are a careful reasoner. Think step by step concisely.
Then on a new line, output exactly: 'Final answer: A' or 'Final answer: B'.
```

LogicalDeduction. LogicalDeduction is a multiple-choice deductive reasoning task. Let $O = \{A, B, C, D\}$ denote the option set (dataset-

dependent). We instruct the model to output the selected option letter in a strict format:

You are a careful reasoner. Think step by step concisely.
Then on a new line, output exactly: ‘Final answer: <option letter>’.

AR-LSAT. AR-LSAT involves multi-step analytical reasoning with multiple-choice answers. We use the same standardized option-letter output format to enable robust evaluation:

You are a careful analytical reasoner. Think step by step concisely.
Then on a new line, output exactly: ‘Final answer: <option letter>’.

GSM8K. For math word problems, we follow the conventional Chain-of-Thought prompting format and enforce a strict numeric final answer:

You are a careful math reasoner. Solve step by step concisely.
Then on a new line, output exactly: ‘Final answer: <number>’.

Remark. Across all datasets, we keep the reasoning instruction intentionally brief and enforce a single-line final answer in a fixed pattern. During evaluation, we extract the prediction by matching the last occurrence of the Final answer: prefix and parsing the subsequent token(s) according to the task type (option letter or number).

A.3 Implementation Details

A.3.1 Model Inference

All experiments are conducted using vLLM (Kwon et al., 2023) as the inference backend to ensure efficient serving of large models and fast hidden-state extraction. Unless otherwise specified, model precision is set to FP16, following the default mixed-precision configuration of vLLM. We use HuggingFace Transformers (Wolf et al., 2020) for model loading, tokenization, and hidden-state access.

A.3.2 Generation Hyperparameters

As shown in Table 2, across all experiments—including cross-domain ICL evaluation, DIN retrieval, and case studies—we use the following decoding configuration:

A.4 Retrieval Methods

This section details the retrieval methods used to select in-context demonstrations for cross-domain

Category	Setting
Temperature	0.0(Greedy)
Max Gen Length	8192 tokens
Random Seed	1-30

Table 2: Decoding setup used throughout all experiments.

ICL, including dense vector retrieval and sparse lexical retrieval (BM25). All retrieval methods operate on the source-domain demonstration pool $\mathcal{D}_S = \{x_i\}_{i=1}^N$ and return the top- k examples for a given target query x_q .

A.4.1 Dense Vector Retrieval

In dense retrieval, each example x is mapped to a continuous embedding vector via an encoder $f_\theta(\cdot)$:

$$\mathbf{h}_x = f_\theta(x) \in \mathbb{R}^d, \quad (6)$$

where f_θ is a pretrained sentence or text encoder. The target query x_q is embedded in the same space as \mathbf{h}_{x_q} .

Retrieval is performed by computing a similarity score between the query embedding and each candidate example. We adopt cosine similarity:

$$\text{sim}_{\text{dense}}(x_q, x_i) = \frac{\mathbf{h}_{x_q}^\top \mathbf{h}_{x_i}}{\|\mathbf{h}_{x_q}\| \|\mathbf{h}_{x_i}\|}. \quad (7)$$

The top- k demonstrations are selected by:

$$\mathcal{R}_k^{\text{dense}}(x_q) = \arg \text{topk}_{x_i \in \mathcal{D}_S} \text{sim}_{\text{dense}}(x_q, x_i). \quad (8)$$

Dense retrieval captures semantic similarity in a continuous representation space, enabling soft matching beyond exact lexical overlap.

A.4.2 BM25 Retrieval

BM25 is a classical sparse retrieval method based on term-level exact matching. Each example x is treated as a bag of tokens. Given a query x_q , the BM25 score between x_q and a candidate example x_i is computed as:

$$\text{sim}_{\text{BM25}}(x_q, x_i) = \frac{\sum_{t \in x_q} \text{IDF}(t) \cdot \frac{f(t, x_i) (k_1 + 1)}{f(t, x_i) + k_1 \left(1 - b + b \cdot \frac{|x_i|}{\text{avgdl}}\right)}}{\quad} \quad (9)$$

where $f(t, x_i)$ denotes the term frequency of token t in x_i , $|x_i|$ is the length of x_i , and avgdl is

the average document length in \mathcal{D}_S . The inverse document frequency is defined as:

$$\text{IDF}(t) = \log \frac{N - n_t + 0.5}{n_t + 0.5}, \quad (10)$$

with n_t being the number of examples containing token t . We follow standard practice and set k_1 and b to fixed constants.

The retrieved set is obtained by:

$$\mathcal{R}_k^{\text{BM25}}(x_q) = \arg \text{topk}_{x_i \in \mathcal{D}_S} \text{sim}_{\text{BM25}}(x_q, x_i). \quad (11)$$

Unlike dense retrieval, BM25 relies on exact lexical overlap and does not use learned representations, providing a strong non-neural baseline.

A.5 TopK+ConE: Data- & Model-Dependent Demonstration Selection

Given a test input x and a demonstration pool \mathcal{D} , we aim to select N in-context demonstrations c . We formalize effective demonstration selection as minimizing the conditional entropy of the test input x under an inference model p_θ :

$$c^* = \arg \min_{c \in \mathcal{C}} H_\theta(x | c), \quad (12)$$

where \mathcal{C} denotes the candidate set of demonstration groups and

$$H_\theta(x | c) = -\mathbb{E}_x [\log p_\theta(x | c)]. \quad (13)$$

Using the chain rule of (cross-)entropy, Eq. (12) can be rewritten as

$$H_\theta(x | c) = H_\theta(x, c) - H_\theta(c), \quad (14)$$

where $H_\theta(x, c)$ is the cross-entropy of the concatenated prompt (demonstrations plus test input) and $H_\theta(c)$ is the cross-entropy of the demonstrations alone, both estimated by the same inference model p_θ . Thus, TopK+ConE ranks candidate demonstrations by the difference $H_\theta(x, c) - H_\theta(c)$.

Select-then-Rerank. Enumerating \mathcal{C} is infeasible, so we employ a two-stage pipeline. First, a data-dependent retriever $r(\cdot)$ (e.g., embedding-based nearest neighbors) selects a candidate set $\mathcal{D}_K(x) \subset \mathcal{D}$ with $|\mathcal{D}_K(x)| = K$:

$$\mathcal{D}_K(x) = \text{TopK}_{d \in \mathcal{D}} s(r(x), r(d)), \quad (15)$$

where $s(\cdot, \cdot)$ is a similarity function. Then we form a small set of candidate demonstration groups

$\mathcal{C}(x) \subseteq \binom{\mathcal{D}_K(x)}{N}$ and rerank each $c \in \mathcal{C}(x)$ by ConE:

$$\begin{aligned} \text{score}(c; x) &= H_\theta(x, c) - H_\theta(c), \\ c^* &= \arg \min_{c \in \mathcal{C}(x)} \text{score}(c; x). \end{aligned} \quad (16)$$

Finally, the prompt $\langle c^*, x \rangle$ is fed into the inference model to produce the prediction \hat{y} .

A.5.1 Usage in ICL

For all experiments, the retrieved demonstrations $\mathcal{R}_k(\cdot)$ are concatenated with the query in a fixed prompt template and fed into the frozen language model. Apart from the retrieval mechanism, all other components of the ICL pipeline are kept identical to ensure fair comparison across retrieval methods.

A.6 Results of BM25 Retrieval

Figures 11 and 12 report cross-domain ICL performance when demonstrations are retrieved using BM25 lexical matching, under the same experimental protocol as the embedding-based retriever in the main text.

Overall, BM25 exhibits substantially weaker and less stable cross-domain transfer compared to dense embedding retrieval. As shown in Figure 11, BM25 rarely achieves consistent improvements over zero-shot baselines across source–target pairs, and negative transfer is more frequent, especially when the source and target domains differ in surface form or vocabulary. This behavior contrasts with embedding-based retrieval, which demonstrates more reliable gains under similar settings.

The performance gap becomes more pronounced as the number of demonstrations increases. Figure 12 shows that BM25 does not benefit from shot scaling in most cross-domain configurations. In several cases, increasing the number of BM25-retrieved demonstrations leads to flat or even degraded performance, indicating that lexical similarity alone is insufficient to identify demonstrations that provide transferable reasoning patterns. This phenomenon is particularly evident for targets requiring abstract logical structures, such as FOLIO and AR-LSAT.

Nevertheless, BM25 is not uniformly ineffective. In cases where source and target domains share strong lexical overlap and similar surface templates (e.g., within closely related logical reasoning datasets), BM25 can occasionally match

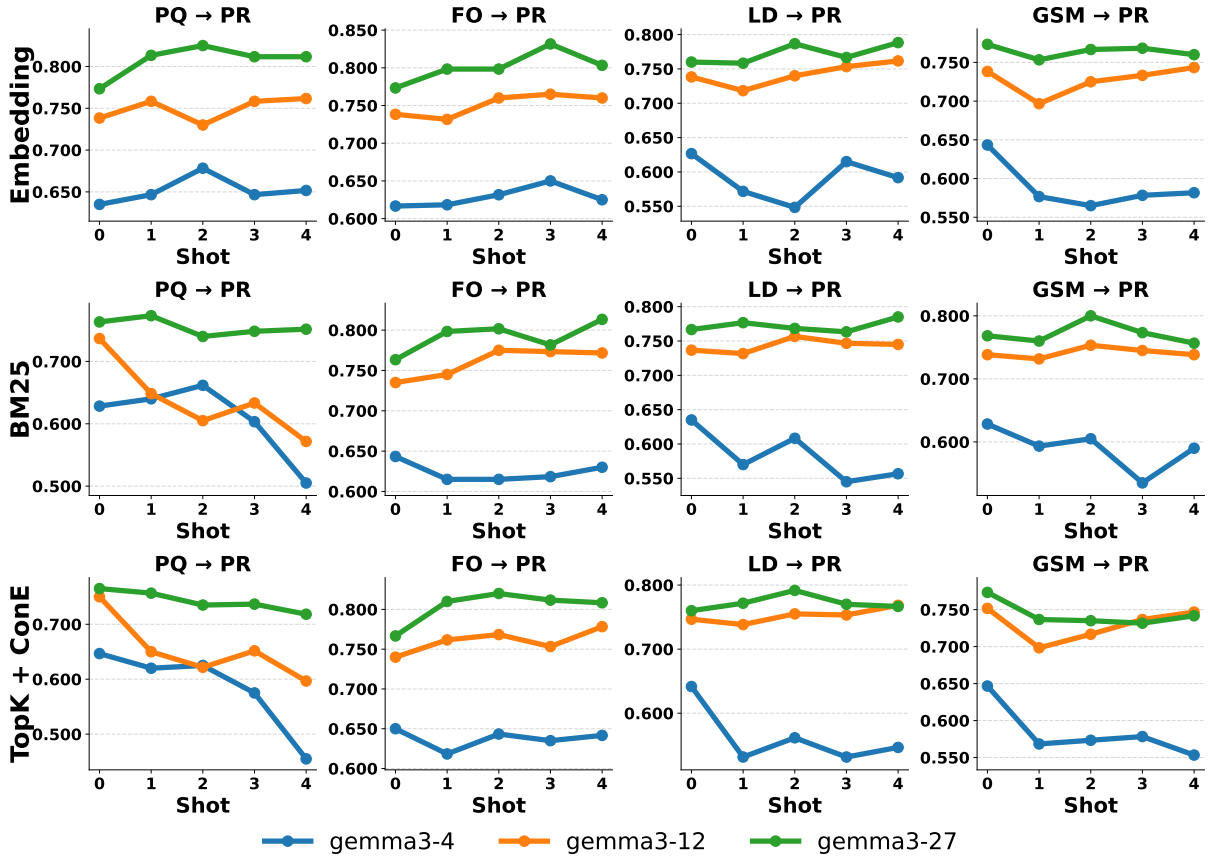


Figure 7: **Shot–performance scaling across cross-domain ICL settings.** Accuracy curves for multiple models and transfer directions show that increasing the number of demonstrations does not yield reliable positive scaling. While structurally aligned transfers exhibit eAR-LSATy saturation, mismatched pairs often experience instability or negative transfer as shot size increases.

or approach the performance of embedding-based retrieval, especially at small shot numbers. This suggests that BM25 may still serve as a competitive baseline when reasoning structures and language realizations are highly aligned.

Taken together, these results highlight a key limitation of purely lexical retrieval for cross-domain ICL. While BM25 can retrieve demonstrations that are textually similar, it often fails to capture the structural compatibility required for effective reasoning transfer. This finding further supports our main conclusion that cross-domain ICL performance depends more critically on structural and difficulty alignment than on surface-level similarity.

A.7 Results of Top-K+ConE Retrieval

A.8 Statistical Significance Tests

This appendix reports the statistical tests used throughout the empirical study and summarizes the corresponding numerical results. Unless otherwise stated, all tests are conducted on paired experimen-

tal outcomes under identical settings (same model, source–target pair, and shot number).

A.8.1 Dense Retrieval vs. BM25 Retrieval (RQ1)

We compare embedding-based dense retrieval and BM25 sparse retrieval using paired Exact Match (EM) scores across all models and cross-domain transfer directions. Since EM differences are not guaranteed to be normally distributed, we primarily adopt the Wilcoxon signed-rank test, supplemented by paired t -tests for reference.

Metric	Value
Mean EM (Dense Retrieval)	0.6658
Mean EM (BM25)	0.6671
Mean Difference (BM25 – Dense)	+0.0013
Wilcoxon statistic	29845.0
Wilcoxon p -value	0.2803
Paired t -test t	1.1458
Paired t -test p -value	0.2526

Table 3: Overall significance test comparing BM25 and dense retrieval.

Overall, BM25 does not yield a statistically

significant improvement over dense retrieval (Wilcoxon $p = 0.28$), with an average gain of only +0.13 percentage points in EM.

Per-model analysis. Table 4 reports significance tests stratified by model.

Model	n	Mean Diff	Wilcoxon p	t -test p
LLaMA3-8B	100	+0.0029	0.818	0.306
Qwen2.5-14B	100	+0.0031	0.093	0.157
Qwen3-4B	100	+0.0053	0.135	0.024
Qwen2.5-32B	145	-0.0009	0.692	0.626
Qwen2.5-7B	100	+0.0005	0.850	0.790

Table 4: Per-model significance tests for BM25 vs. dense retrieval.

Only the smallest model (Qwen3-4B) shows weak sensitivity under the t -test, while all other models exhibit no significant differences.

Per-direction analysis. Certain source–target transfer directions exhibit statistically significant differences (Table 5), suggesting that dense retrieval is more effective when semantic distance is large and reasoning structure is complex.

Source \rightarrow Target	Mean Diff	Wilcoxon p	t -test p
FOL \rightarrow GSM	-0.0039	0.0030	0.0035
LD \rightarrow FOL	-0.0108	0.0089	0.0071
PQA \rightarrow PW	-0.0497	0.0679	0.0333
PW \rightarrow GSM	-0.0055	0.0254	0.0179
PW \rightarrow LD	+0.0407	0.0625	0.0021

Table 5: Per-direction significance tests for BM25 vs. dense retrieval.

A.8.2 Model Size and Monotonicity (RQ2)

We further examine the relationship between model size and few-shot monotonicity.

Metric	Coefficient	p -value
Pearson r	0.2070	1.56×10^{-4}
Spearman ρ	0.2079	1.46×10^{-4}
Kendall τ	0.1510	1.50×10^{-4}

Table 6: Correlation between model size and few-shot monotonicity.

All correlations are significantly positive, supporting the existence of a model-size-dependent “example absorption” threshold for effective cross-domain few-shot learning.

A.8.3 Few-shot Monotonicity (RQ3)

We analyze the relationship between task complexity (defined as the product of source and target option counts) and few-shot monotonicity, measured

Correlation Metric	Coefficient	p -value
Pearson r	-0.2336	1.87×10^{-5}
Spearman ρ	-0.3864	3.72×10^{-13}
Kendall τ	-0.2808	1.22×10^{-12}

Table 7: Overall correlation between task complexity and few-shot monotonicity.

by the Spearman correlation between shot number and EM.

All three correlation coefficients are significantly negative, indicating that higher task complexity is associated with less reliable few-shot scaling.

A.9 Full Results

Tables 10–13 report the complete cross-domain ICL results across all evaluated models, source–target domain pairs, retrieval methods, and shot settings. These tables serve as comprehensive supplementary evidence for the main empirical observations discussed in the paper, and are intended to demonstrate that our conclusions are not driven by a small subset of tasks or configurations.

Non-monotonic shot scaling. Across Tables 10–13, increasing the number of demonstrations does not consistently lead to performance gains in cross-domain ICL. While moderate numbers of shots can be beneficial in some settings, higher shot counts frequently exhibit diminishing returns or even performance degradation. This non-monotonic behavior is especially pronounced when the source and target domains differ substantially in reasoning structure or difficulty, supporting our claim that cross-domain ICL is not governed by simple shot scaling laws.

Strong dependence on model capacity. The full results further reveal a clear interaction between model size and demonstration effectiveness. Smaller models tend to show high variance and are more susceptible to negative transfer when additional demonstrations are introduced, whereas larger models exhibit more stable improvements and are better able to absorb cross-domain demonstrations. This pattern consistently appears across multiple targets and retrieval settings, indicating the presence of a model-dependent example absorption threshold.

Target-domain sensitivity. Another salient pattern observed in Tables 10–13 is the strong dependence of cross-domain ICL gains on the target domain. Certain target tasks benefit more reliably

1053 from retrieved demonstrations, while others show
1054 limited or inconsistent improvements regardless
1055 of the source domain or shot count. This target-
1056 specific sensitivity suggests that the effectiveness
1057 of demonstrations is constrained by the reasoning
1058 requirements of the target task, rather than by the
1059 availability of semantically similar examples alone.

Retrieval method comparison. Comparing
1060 dense retrieval and BM25 across the full results,
1061 we observe that neither method universally dom-
1062 inates the other. Dense retrieval often provides
1063 stronger gains when semantic alignment between
1064 source and target exists, while BM25 remains com-
1065 petitive or more stable in settings where lexical
1066 overlap is informative. The variability across Ta-
1067 bles 10–13 highlights that retrieval quality alone
1068 cannot guarantee effective cross-domain transfer
1069 without considering task structure and difficulty.

Overall, the comprehensive results in Tables 10–
1071 13 reinforce our central findings: cross-domain ICL
1072 performance is highly heterogeneous, sensitive to
1073 model capacity and target-domain characteristics,
1074 and cannot be reliably improved by increasing the
1075 number of demonstrations or by relying on a single
1076 retrieval strategy.
1077

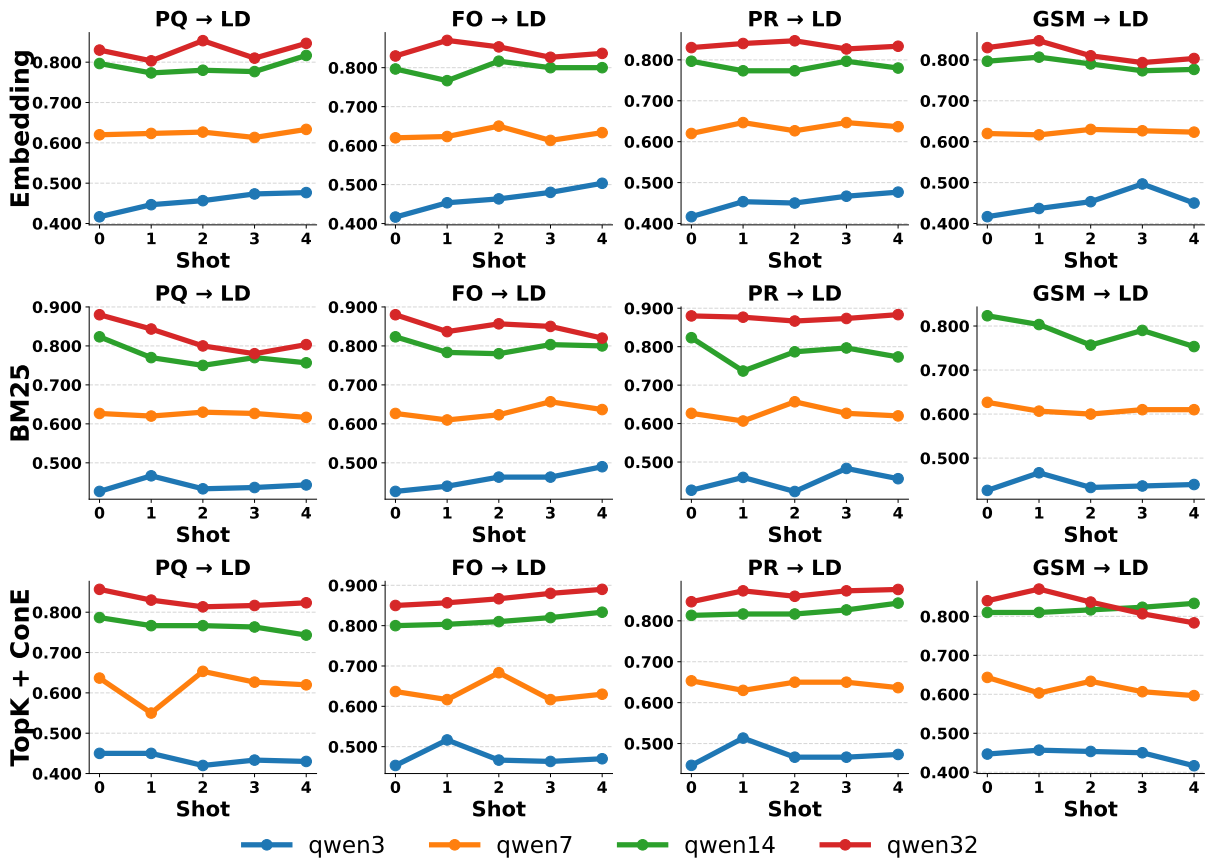


Figure 8: **Shot-performance scaling across cross-domain ICL settings.** Accuracy curves for multiple models and transfer directions show that increasing the number of demonstrations does not yield reliable positive scaling. While structurally aligned transfers exhibit eAR-LSATy saturation, mismatched pairs often experience instability or negative transfer as shot size increases.

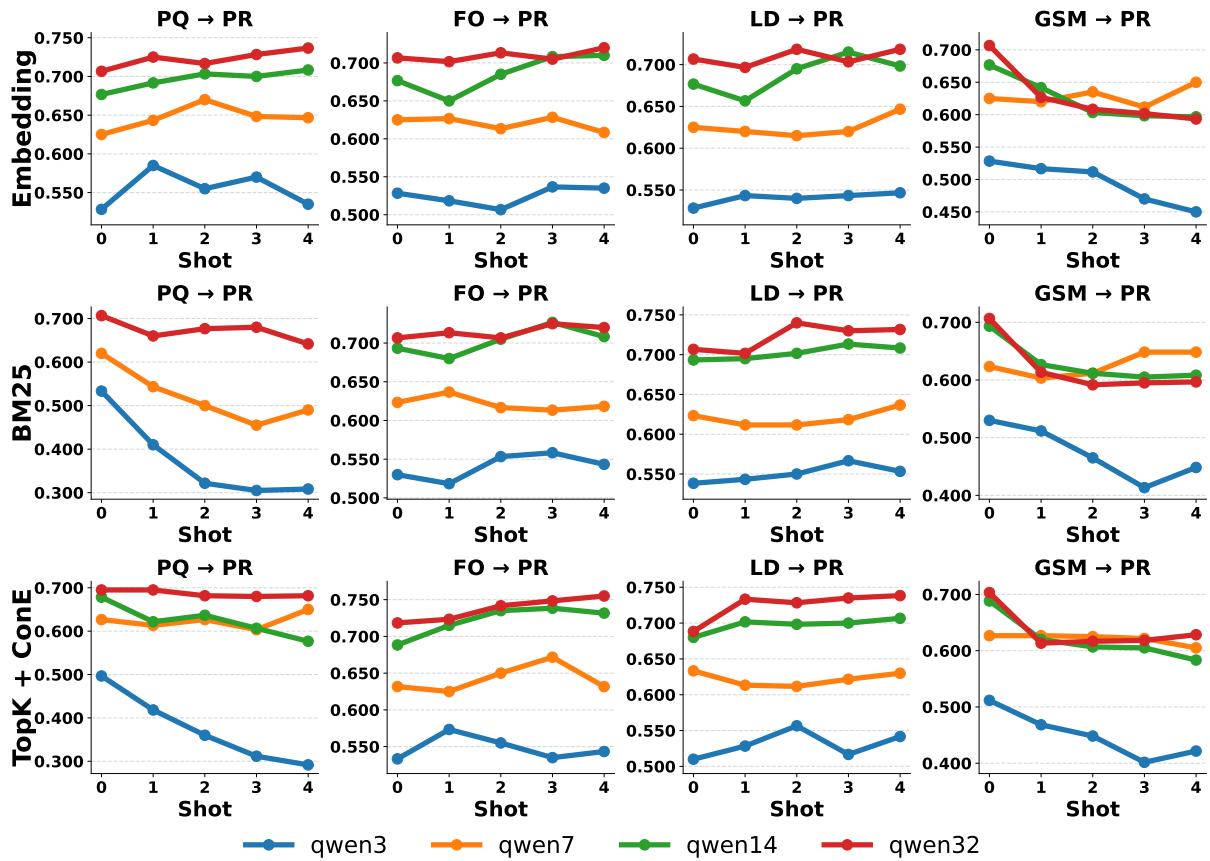


Figure 9: **Shot–performance scaling across cross-domain ICL settings.** Accuracy curves for multiple models and transfer directions show that increasing the number of demonstrations does not yield reliable positive scaling. While structurally aligned transfers exhibit eAR-LSATy saturation, mismatched pairs often experience instability or negative transfer as shot size increases.

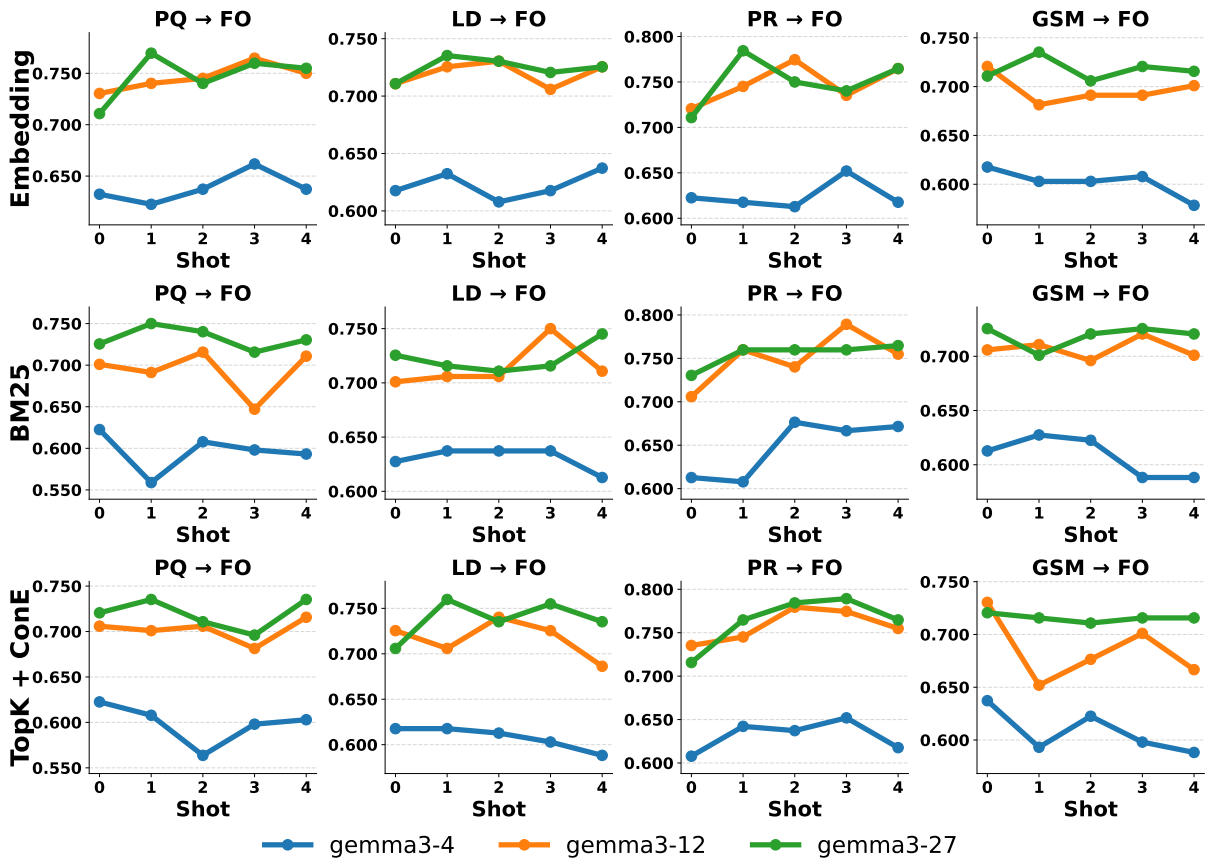


Figure 10: **Shot–performance scaling across cross-domain ICL settings.** Accuracy curves for multiple models and transfer directions show that increasing the number of demonstrations does not yield reliable positive scaling. While structurally aligned transfers exhibit eAR-LSATy saturation, mismatched pairs often experience instability or negative transfer as shot size increases.

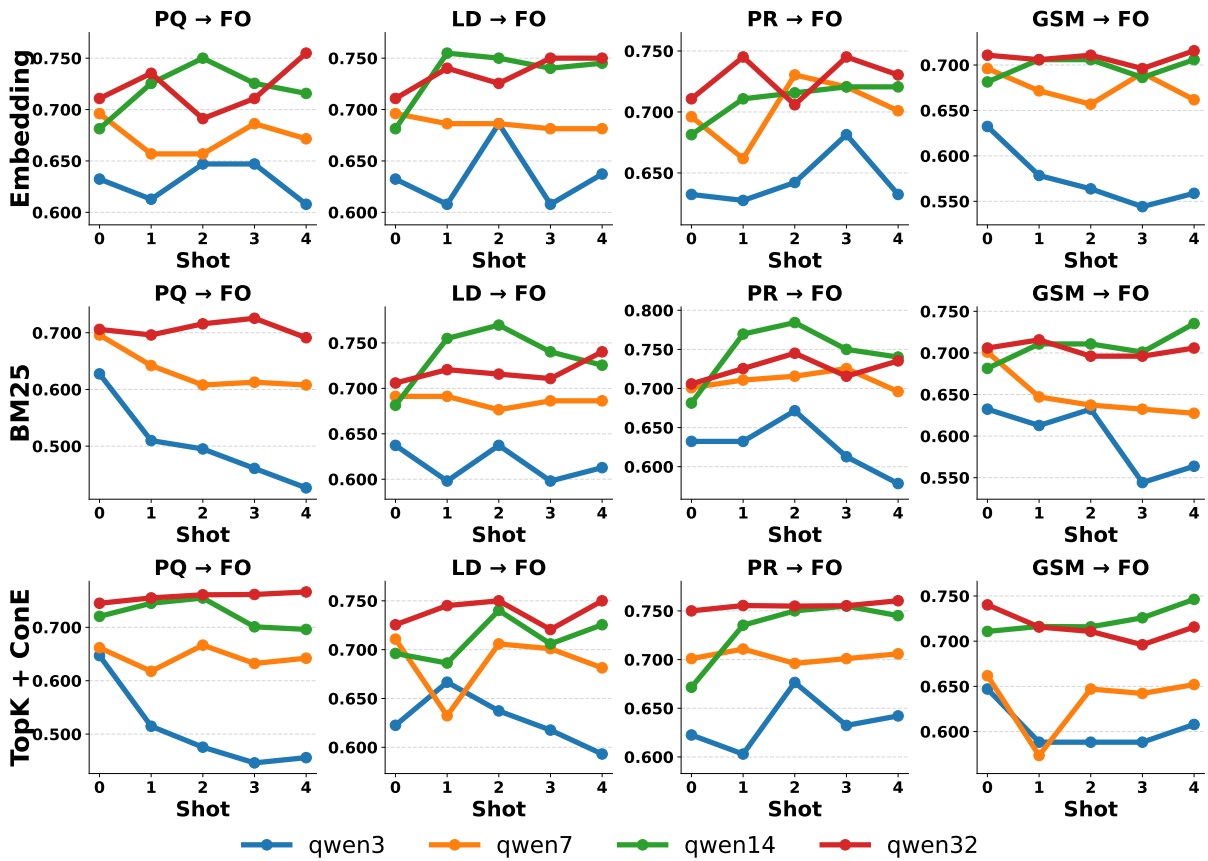


Figure 11: **Shot–performance scaling across cross-domain ICL settings.** Accuracy curves for multiple models and transfer directions show that increasing the number of demonstrations does not yield reliable positive scaling. While structurally aligned transfers exhibit eAR-LSATy saturation, mismatched pairs often experience instability or negative transfer as shot size increases.

Model	Source	AR-LSAT					FOLIO					LogicalDeduction					ProntoQA					ProofWriter					gsm8k					
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	
qwen3	FOLIO	0.235	0.252	0.178	0.257	0.209	0.632	0.593	0.627	0.642	0.623	0.417	0.453	0.463	0.480	0.503	0.858	0.846	0.846	0.844	0.844	0.844	0.528	0.518	0.507	0.537	0.535	0.737	0.726	0.682	0.659	0.645
	LogicalDeduction	0.235	0.191	0.235	0.209	0.265	0.632	0.608	0.686	0.608	0.637	0.417	0.487	0.513	0.470	0.450	0.858	0.856	0.820	0.832	0.832	0.832	0.528	0.543	0.540	0.543	0.547	0.737	0.726	0.717	0.699	0.663
	ProntoQA	0.235	0.187	0.187	-	-	0.632	0.613	0.647	0.647	0.608	0.417	0.447	0.457	0.473	0.473	0.858	0.854	0.906	0.896	0.896	0.528	0.585	0.555	0.570	0.545	0.737	0.733	0.714	0.723	0.710	
	ProofWriter	0.235	0.213	0.226	0.196	0.187	0.632	0.627	0.642	0.681	0.632	0.417	0.453	0.450	0.467	0.477	0.858	0.856	0.872	0.846	0.860	0.528	0.528	0.528	0.513	0.510	0.737	0.742	0.689	0.651	0.624	
	gsm8k	0.235	0.174	0.226	0.217	0.230	0.632	0.578	0.564	0.544	0.559	0.417	0.437	0.453	0.497	0.450	0.858	0.752	0.734	0.592	0.574	0.528	0.517	0.512	0.470	0.450	0.737	0.753	0.729	0.721	0.665	
qwen7	FOLIO	0.257	0.270	0.230	0.222	0.217	0.696	0.676	0.696	0.711	0.701	0.620	0.623	0.650	0.613	0.968	0.958	0.964	0.968	0.978	0.968	0.625	0.627	0.613	0.628	0.608	0.890	0.885	0.867	0.867	0.828	
	LogicalDeduction	0.257	0.243	0.257	0.257	0.248	0.696	0.686	0.686	0.681	0.681	0.620	0.647	0.683	0.640	0.968	0.980	0.980	0.980	0.988	0.988	0.625	0.620	0.615	0.620	0.647	0.890	0.880	0.876	0.880	0.862	
	ProntoQA	0.257	0.226	0.200	0.222	0.226	0.696	0.657	0.657	0.686	0.672	0.620	0.623	0.627	0.613	0.968	0.944	0.950	0.972	0.964	0.964	0.625	0.643	0.670	0.648	0.647	0.890	0.864	0.877	0.867	0.859	
	ProofWriter	0.257	0.230	0.287	0.213	0.283	0.696	0.662	0.730	0.721	0.701	0.620	0.647	0.627	0.647	0.968	0.970	0.988	0.984	0.992	0.984	0.625	0.627	0.655	0.630	0.658	0.890	0.860	0.867	0.839	0.826	
	gsm8k	0.257	0.226	0.217	0.213	0.230	0.696	0.672	0.657	0.691	0.662	0.617	0.630	0.627	0.623	0.968	0.942	0.924	0.934	0.924	0.934	0.625	0.620	0.635	0.612	0.650	0.890	0.882	0.889	0.885	0.877	
qwen14	FOLIO	0.348	0.274	0.335	0.374	0.322	0.681	0.750	0.745	0.765	0.760	0.797	0.817	0.800	0.800	0.948	0.978	0.978	0.970	0.978	0.978	0.677	0.650	0.685	0.708	0.710	0.932	0.927	0.927	0.929	0.930	
	LogicalDeduction	0.348	0.296	0.317	0.265	0.317	0.681	0.755	0.750	0.740	0.745	0.797	0.810	0.807	0.800	0.948	0.978	0.956	0.968	0.974	0.677	0.657	0.698	0.715	0.698	0.932	0.929	0.926	0.932	0.925		
	ProntoQA	0.348	0.317	0.357	0.326	0.352	0.681	0.725	0.750	0.725	0.716	0.797	0.773	0.780	0.777	0.948	0.972	0.978	0.982	0.984	0.677	0.692	0.703	0.700	0.708	0.932	0.929	0.927	0.929	0.930		
	ProofWriter	0.348	0.287	0.330	0.365	0.348	0.681	0.711	0.716	0.721	0.721	0.797	0.773	0.773	0.797	0.780	0.968	0.982	0.980	0.976	0.677	0.738	0.752	0.762	0.737	0.932	0.932	0.929	0.936	0.932		
	gsm8k	0.348	0.326	0.330	0.300	0.296	0.681	0.706	0.686	0.706	0.797	0.797	0.807	0.790	0.773	0.942	0.920	0.892	0.896	0.677	0.642	0.603	0.597	0.932	0.922	0.924	0.923	0.924	0.923	0.923	0.926	
qwen32	FOLIO	0.322	0.335	0.278	0.365	0.343	0.711	0.740	0.745	0.735	0.745	0.830	0.870	0.853	0.827	0.837	1.000	1.000	1.000	1.000	1.000	0.707	0.702	0.713	0.705	0.720	0.883	0.900	0.907	0.900	0.896	
	LogicalDeduction	0.322	0.343	0.335	0.283	0.313	0.711	0.740	0.725	0.750	0.750	0.830	0.867	0.860	0.853	0.820	1.000	0.998	0.998	1.000	1.000	0.707	0.697	0.718	0.703	0.718	0.883	0.902	0.917	0.914	0.909	
	ProntoQA	0.322	0.343	0.313	0.339	0.383	0.711	0.735	0.691	0.711	0.755	0.830	0.803	0.853	0.810	0.847	1.000	0.998	0.998	1.000	0.998	0.707	0.725	0.717	0.728	0.737	0.883	0.903	0.908	0.910	0.899	
	ProofWriter	0.322	0.343	0.322	0.343	0.352	0.711	0.745	0.706	0.745	0.730	0.830	0.840	0.847	0.827	0.833	1.000	0.998	1.000	0.998	1.000	0.707	0.732	0.735	0.753	0.757	0.883	0.889	0.910	0.885	0.860	
	gsm8k	0.322	0.348	0.322	0.361	0.352	0.711	0.706	0.711	0.696	0.716	0.830	0.847	0.810	0.793	0.803	1.000	0.992	0.984	0.984	0.982	0.707	0.627	0.608	0.602	0.593	0.883	0.910	0.920	0.917	0.920	
gemma3-4	FOLIO	0.252	0.270	0.265	0.243	0.222	0.652	0.623	0.652	0.618	0.642	0.610	0.587	0.560	0.587	0.583	0.920	0.930	0.924	0.920	0.910	0.617	0.618	0.632	0.650	0.625	0.864	0.850	0.845	0.837	0.831	
	LogicalDeduction	0.278	0.235	0.226	0.248	0.261	0.618	0.632	0.608	0.618	0.637	0.613	0.650	0.637	0.637	0.573	0.930	0.880	0.892	0.906	0.898	0.627	0.572	0.548	0.615	0.592	0.859	0.843	0.843	0.845	0.840	
	ProntoQA	0.283	0.257	0.222	0.252	0.252	0.632	0.623	0.637	0.662	0.637	0.563	0.573	0.577	0.583	0.580	0.910	0.934	0.946	0.934	0.954	0.635	0.647	0.678	0.647	0.652	0.861	0.848	0.845	0.845	0.841	
	ProofWriter	0.252	0.239	0.261	0.252	0.257	0.623	0.618	0.613	0.652	0.618	0.607	0.547	0.553	0.580	0.610	0.928	0.956	0.930	0.930	0.914	0.638	0.673	0.648	0.635	0.623	0.860	0.832	0.842	0.845	0.835	
	gsm8k	0.287	0.252	0.265	0.270	0.274	0.618	0.603	0.603	0.608	0.578	0.627	0.630	0.610	0.593	0.550	0.926	0.866	0.840	0.808	0.782	0.643	0.577	0.565	0.578	0.582	0.860	0.831	0.819	0.786	0.754	
gemma3-12	FOLIO	0.300	0.335	0.335	0.304	0.317	0.725	0.725	0.770	0.740	0.770	0.733	0.773	0.780	0.760	0.733	0.982	0.978	0.980	0.988	0.976	0.738	0.732	0.760	0.765	0.760	0.920	0.908	0.905	0.892	0.899	
	LogicalDeduction	0.300	0.248	0.270	0.304	0.309	0.711	0.725	0.730	0.706	0.725	0.747	0.847	0.827	0.837	0.807	0.982	0.980	0.980	0.982	0.978	0.738	0.718	0.740	0.753	0.762	0.918	0.917	0.907	0.900	0.899	
	ProntoQA	0.330	0.274	0.257	0.304	0.309	0.730	0.740	0.745	0.765	0.750	0.730	0.743	0.787	0.747	0.767	0.982	-	0.990	0.996	0.996	0.738	0.758	0.730	0.758	0.762	0.920	0.910	0.912	0.900	0.899	
	ProofWriter	0.330	0.296	0.261	0.326	0.270	0.721	0.745	0.775	0.735	0.765	0.730	0.763	0.773	0.760	0.783	0.982	0.986	0.992	0.986	0.992	0.738	0.752	0.763	0.792	0.790	0.921	0.910	0.905	0.898	0.895	
	gsm8k	0.330	0.322	0.291	0.322	0.313	0.721	0.681	0.691	0.691	0.701	0.730	0.747	0.747	0.750	0.733	0.982	0.980	0.976	0.978	0.978	0.738	0.697	0.725	0.743	0.920	0.871	0.608	0.624	0.569		
gemma3-27	FOLIO	0.348	0.370	0.361	0.357	0.365	0.711	0.750	0.725	0.711	0.770	0.777	0.860	0.827	0.803	0.982	0.988	0.988	0.984	0.990	0.773	0.798	0.798	0.832	0.803	0.932	0.933	0.931	0.934	0.924		
	LogicalDeduction	0.400	0.374	0.426	0.400	0.439	0.711	0.735	0.730	0.721	0.725	0.777	0.880	0.883	0.890	0.897	0.984	0.984	0.982	0.984	0.982	0.760	0.758	0.787	0.767	0.788	0.932	0.926	0.922	0.922	0.923	
	ProntoQA	0.400	0.391	0.378	0.357	0.383	0.711	0.770	0.740	0.760	0.755	0.777	0.833	0.870	0.870	0.880	0.982	0.996	0.992	0.998	0.994	0.773	0.813	0.825	0.812	0.812	0.932	0.933	0.925	0.918	0.923	
	ProofWriter	0.396	0.387	0.400	0.378	0.374	0.711	0.784	0.750	0.740	0.765																					

Model	Source	AR-LSAT					FOLIO					LogicalDeduction					ProntoQA					ProofWriter					gsm8k				
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
qwen3-4	FOLIO	0.696	0.657	0.643	0.635	0.600	0.804	0.809	0.794	0.824	0.814	0.963	0.963	0.943	0.943	0.933	0.998	0.996	0.994	0.992	0.992	0.827	0.807	0.808	0.815	0.818	0.916	0.901	0.905	0.910	0.907
	LogicalDeduction	0.696	0.670	0.678	0.622	0.652	0.804	0.789	0.799	0.794	0.779	0.963	0.957	0.960	0.967	0.940	1.000	1.000	0.998	1.000	0.998	1.000	0.827	0.795	0.820	0.823	0.798	0.917	0.906	0.907	0.913
	ProntoQA	0.696	0.674	0.674	0.652	0.613	0.804	0.809	0.809	0.809	0.799	0.963	0.940	0.950	0.933	0.950	0.998	0.990	0.996	0.992	0.992	0.827	0.797	0.793	0.788	0.803	0.917	0.910	0.900	0.889	0.895
	ProofWriter	0.696	0.665	0.596	0.604	0.548	0.804	0.779	0.809	0.804	0.804	0.963	0.930	0.937	0.910	0.923	0.998	0.992	0.994	1.000	1.000	1.000	0.827	0.805	0.815	0.817	0.800	0.917	0.894	0.902	0.910
	gsm8k	0.696	0.678	0.652	0.709	0.661	0.804	0.779	0.819	0.799	0.789	0.963	0.963	0.960	0.967	0.960	0.998	0.992	0.998	0.992	0.994	0.998	0.827	0.810	0.807	0.823	0.815	0.917	0.920	0.912	0.919
qwen3-8	FOLIO	0.643	0.687	0.665	0.613	0.600	0.784	0.809	0.794	0.804	0.833	0.920	0.980	0.940	0.960	1.000	1.000	1.000	1.000	1.000	0.882	0.858	0.845	0.865	0.850	0.929	0.920	0.917	0.914	0.914	
	LogicalDeduction	0.643	0.661	0.674	0.630	0.661	0.784	0.794	0.804	0.804	0.775	0.920	0.983	0.980	0.967	1.000	1.000	1.000	0.996	0.998	1.000	0.882	0.860	0.860	0.848	0.847	0.929	0.932	0.926	0.920	0.916
	ProntoQA	0.643	0.657	0.639	0.678	0.613	0.784	0.828	0.814	0.809	0.819	0.920	0.950	0.930	0.947	1.000	0.984	0.996	0.998	0.994	0.994	0.882	0.840	0.830	0.848	0.857	0.929	0.917	0.928	0.920	0.927
	ProofWriter	0.643	0.652	0.657	0.600	0.561	0.784	0.828	0.824	0.814	0.843	0.920	0.960	0.950	0.960	1.000	0.998	1.000	0.998	1.000	0.998	0.882	0.857	0.872	0.860	0.842	0.929	0.923	0.920	0.917	0.915
	gsm8k	0.643	0.704	0.709	0.661	0.700	0.784	0.814	0.784	0.804	0.833	0.920	0.960	0.940	0.967	0.953	1.000	0.996	0.998	0.996	1.000	0.882	0.865	0.868	0.865	0.865	0.929	0.934	0.936	0.940	
qwen3-14	FOLIO	0.774	0.752	0.743	0.704	0.670	0.824	0.848	0.824	0.814	0.814	0.990	0.993	0.983	0.977	0.993	0.990	0.992	0.994	0.996	0.998	0.840	0.842	0.827	0.813	0.833	0.878	0.915	0.914	0.911	0.913
	LogicalDeduction	0.774	0.757	0.752	0.717	0.713	0.824	0.848	0.843	0.828	0.819	0.990	0.997	0.990	0.990	0.980	0.990	0.994	0.984	0.984	0.988	0.840	0.822	0.815	0.812	0.810	0.878	0.914	0.910	0.908	0.908
	ProntoQA	0.774	0.739	0.674	0.652	0.657	0.824	0.824	0.824	0.848	0.833	0.990	0.993	0.967	0.987	0.990	1.000	0.994	0.996	0.992	1.000	0.840	0.825	0.802	0.822	0.827	0.878	0.915	0.914	0.913	0.915
	ProofWriter	0.774	0.743	0.687	0.657	0.643	0.824	0.824	0.833	0.824	0.824	0.990	0.983	0.983	0.980	0.993	0.990	0.992	0.992	0.992	0.998	0.840	0.843	0.825	0.853	0.857	0.878	0.913	0.914	0.912	0.915
	gsm8k	0.774	0.752	0.757	0.757	0.739	0.824	0.809	0.824	0.809	0.814	0.990	0.990	0.987	0.980	0.993	0.990	0.984	0.974	0.964	0.964	0.840	0.832	0.823	0.832	0.832	0.878	0.917	0.924	0.927	0.930
qwen3-32	FOLIO	0.791	0.783	0.717	0.739	0.687	0.804	0.789	0.814	0.799	0.804	0.903	0.877	0.843	0.823	0.913	1.000	1.000	1.000	1.000	0.838	0.855	0.843	0.853	0.853	0.930	0.942	0.938	0.933	0.932	
	LogicalDeduction	0.778	0.757	0.748	0.730	0.726	0.784	0.789	0.814	0.794	0.779	0.860	0.930	0.960	0.957	0.943	1.000	1.000	0.998	0.998	0.860	0.850	0.818	0.838	0.828	0.931	0.937	0.930	0.933	0.936	
	ProntoQA	0.770	0.717	0.730	0.726	0.735	0.775	0.804	0.794	0.794	0.809	0.907	0.903	0.833	0.827	0.723	1.000	1.000	0.996	0.994	0.992	0.843	0.842	0.833	0.838	0.845	0.929	0.936	0.939	0.929	0.932
	ProofWriter	0.752	0.700	-	-	-	0.799	0.770	0.775	0.804	0.814	0.880	0.920	0.920	0.757	0.750	1.000	1.000	1.000	1.000	1.000	0.827	0.877	0.860	0.842	0.863	0.929	0.937	0.936	0.937	0.937
	gsm8k	0.757	0.735	0.748	0.739	0.748	0.775	0.770	0.799	0.789	0.789	0.887	0.770	0.730	0.647	0.650	1.000	1.000	1.000	1.000	1.000	0.843	0.833	0.855	0.837	0.936	0.944	0.945	0.942	0.942	
llama3-8	FOLIO	0.287	0.230	0.200	0.252	0.235	0.593	0.608	0.598	0.574	0.627	0.547	0.557	0.527	0.490	0.480	0.794	0.782	0.798	0.794	0.800	0.448	0.528	0.537	0.518	0.552	0.616	0.588	0.576	0.561	
	LogicalDeduction	0.287	0.278	0.143	0.191	0.252	0.593	0.608	0.593	0.574	0.608	0.547	0.597	0.580	0.580	0.523	0.794	0.840	0.846	0.814	0.850	0.448	0.493	0.527	0.520	0.497	0.616	0.597	0.584	0.580	
	ProntoQA	0.287	0.217	0.209	0.209	0.204	0.593	0.569	0.603	0.623	0.623	0.547	0.467	0.463	0.530	0.550	0.794	0.860	0.892	0.920	0.942	0.448	0.568	0.563	0.543	0.570	0.616	0.598	0.597	0.590	
	ProofWriter	0.287	0.257	0.209	0.243	0.248	0.593	0.618	0.618	0.574	0.598	0.547	0.457	0.477	0.507	0.497	0.794	0.736	0.774	0.780	0.794	0.448	0.538	0.597	0.598	0.560	0.616	0.582	0.587	0.558	
	gsm8k	0.287	0.230	0.222	0.204	0.222	0.593	0.569	0.544	0.520	0.515	0.547	0.490	0.463	0.433	0.463	0.794	0.780	0.782	0.784	0.780	0.448	0.498	0.452	0.447	0.490	0.616	0.660	0.688	0.707	

Table 9: Embedding 2

Model	Source	AR-LSAT					FOLIO					LogicalDeduction					ProntoQA					ProofWriter					gsm8k					
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	
qwen3	FOLIO	0.257	0.196	0.283	0.217	0.200	0.632	0.632	0.623	0.637	0.637	0.427	0.440	0.463	0.463	0.490	0.868	0.828	0.830	0.846	0.844	0.844	0.530	0.518	0.553	0.558	0.543	0.737	0.726	0.675	0.644	0.637
	LogicalDeduction	0.235	0.217	0.235	0.209	0.243	0.637	0.598	0.613	0.427	0.453	0.477	0.497	0.437	0.437	0.850	0.842	0.838	0.814	0.844	0.838	0.814	0.844	0.538	0.543	0.550	0.729	0.732	0.712	0.692	0.677	
	ProntoQA	0.248	0.165	0.157	0.174	0.178	0.627	0.510	0.495	0.461	0.426	0.427	0.467	0.433	0.437	0.854	0.742	0.758	0.774	0.533	0.410	0.322	0.305	0.308	0.307	0.308	0.737	0.702	0.718	0.738	0.734	
	ProofWriter	0.257	0.209	0.204	0.235	0.213	0.632	0.632	0.672	0.613	0.578	0.427	0.460	0.423	0.483	0.457	0.868	0.864	0.848	0.848	0.864	0.864	0.530	0.517	0.552	0.563	0.527	0.737	0.719	0.684	0.623	0.602
	gsm8k	0.257	0.226	0.196	0.222	0.157	0.632	0.613	0.632	0.544	0.564	0.427	0.467	0.433	0.437	0.440	0.868	0.784	0.708	0.600	0.556	0.530	0.512	0.465	0.413	0.448	0.737	0.728	0.757	0.710	0.650	
qwen7	FOLIO	0.252	0.287	0.226	0.222	0.235	0.701	0.696	0.676	0.745	0.735	0.627	0.610	0.623	0.657	0.968	0.968	0.970	0.976	0.986	0.986	0.623	0.637	0.617	0.613	0.618	0.887	0.881	0.880	0.856	0.829	
	LogicalDeduction	0.252	0.257	0.252	0.291	0.278	0.691	0.676	0.686	0.627	0.573	0.580	0.623	0.587	0.707	0.972	0.966	0.966	0.986	0.986	0.986	0.623	0.612	0.612	0.618	0.637	0.889	0.881	0.880	0.878	0.867	
	ProntoQA	0.270	0.252	0.204	0.274	0.235	0.696	0.642	0.608	0.613	0.608	0.627	0.620	0.630	0.627	0.617	0.970	0.828	0.842	0.834	0.852	0.620	0.543	0.500	0.455	0.490	0.888	0.848	0.880	0.875	0.877	
	ProofWriter	0.252	0.243	0.222	0.265	0.278	0.701	0.711	0.716	0.725	0.696	0.627	0.607	0.657	0.627	0.620	0.968	0.964	0.986	0.984	0.976	0.623	0.662	0.635	0.668	0.665	0.887	0.860	0.855	0.852	0.783	
	gsm8k	0.252	0.235	0.239	0.239	0.248	0.701	0.647	0.637	0.632	0.627	0.607	0.600	0.610	0.610	0.914	0.914	0.924	0.916	0.914	0.924	0.623	0.603	0.612	0.648	0.648	0.887	0.882	0.884	0.884	0.863	
qwen14	FOLIO	0.317	0.313	0.304	0.300	0.348	0.681	0.755	0.716	0.730	0.730	0.823	0.783	0.780	0.803	0.800	0.946	0.968	0.972	0.964	0.972	0.680	0.705	0.727	0.708	0.713	0.708	0.932	0.927	0.921	0.923	0.923
	LogicalDeduction	0.317	0.296	0.352	0.322	0.278	0.681	0.755	0.770	0.740	0.725	0.823	0.740	0.750	0.780	0.780	0.944	0.992	0.974	0.974	0.693	0.695	0.702	0.713	0.708	0.932	0.930	0.929	0.927	0.936	0.936	
	ProntoQA	—	—	—	—	—	—	—	—	—	—	0.823	0.770	0.750	0.770	0.757	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	ProofWriter	0.317	0.322	0.309	0.374	0.317	0.681	0.770	0.784	0.750	0.740	0.823	0.737	0.787	0.797	0.773	0.946	0.984	0.996	0.984	0.984	0.693	0.752	0.763	0.743	0.773	0.932	0.927	0.928	0.937	0.935	
	gsm8k	0.317	0.313	0.322	0.326	0.317	0.681	0.711	0.711	0.701	0.735	0.823	0.757	0.797	0.773	0.946	0.896	0.902	0.693	0.627	0.612	0.605	0.608	0.932	0.932	0.923	0.923	0.925	0.925	0.919	0.919	0.919
qwen32	FOLIO	0.326	0.343	0.313	0.309	0.339	0.706	0.725	0.740	0.770	0.735	0.880	0.837	0.857	0.850	0.820	1.000	0.998	1.000	1.000	1.000	1.000	0.707	0.713	0.707	0.725	0.720	0.882	0.898	0.897	0.894	0.892
	LogicalDeduction	0.326	0.322	0.296	0.300	0.343	0.706	0.721	0.716	0.711	0.740	0.880	0.843	0.833	0.873	0.863	1.000	1.000	1.000	0.998	0.707	0.702	0.740	0.730	0.732	0.882	0.904	0.917	0.909	0.914	0.914	
	ProntoQA	0.326	0.343	0.317	0.361	0.400	0.706	0.696	0.716	0.725	0.691	0.880	0.843	0.800	0.780	0.803	1.000	0.978	0.990	0.994	0.996	0.707	0.660	0.677	0.680	0.642	0.882	0.892	0.905	0.907	0.921	0.921
	ProofWriter	0.326	0.357	0.357	0.374	0.317	0.706	0.725	0.745	0.716	0.735	0.880	0.877	0.867	0.873	0.883	1.000	0.998	1.000	1.000	1.000	0.707	0.707	0.735	0.767	0.768	0.882	0.889	0.901	0.870	0.843	0.843
	gsm8k	0.326	0.357	0.317	0.352	0.326	0.706	0.716	0.696	0.696	0.706	—	—	—	—	—	1.000	0.994	0.986	0.994	0.984	0.707	0.613	0.592	0.595	0.597	0.882	0.915	0.922	0.924	0.919	0.919
gemma3-4	FOLIO	0.261	0.248	0.230	0.235	0.213	0.598	0.603	0.603	0.603	0.593	0.686	0.577	0.410	0.397	0.413	0.918	0.862	0.916	0.932	0.922	0.643	0.615	0.618	0.630	0.862	0.836	0.840	0.836	0.831	0.831	
	LogicalDeduction	0.261	0.270	0.235	0.257	0.235	0.627	0.637	0.637	0.637	0.613	0.590	0.357	0.370	0.423	0.387	0.912	0.590	0.638	0.654	0.644	0.635	0.570	0.608	0.545	0.557	0.863	0.841	0.830	0.838	0.836	
	ProntoQA	—	—	—	—	—	—	—	—	—	—	0.623	0.559	0.608	0.598	0.593	0.904	0.846	0.822	0.806	0.806	0.628	0.640	0.662	0.603	0.505	—	—	—	—	—	
	ProofWriter	0.252	0.222	0.278	0.287	0.274	0.613	0.608	0.676	0.667	0.672	0.563	0.397	0.383	0.413	0.400	0.916	0.944	0.936	0.952	0.926	0.637	0.680	0.648	0.653	0.657	0.863	0.854	0.849	0.857	0.846	
	gsm8k	0.257	0.261	0.222	0.257	0.235	0.613	0.627	0.623	0.588	0.588	0.587	0.627	0.487	0.397	0.447	0.924	0.866	0.840	0.810	0.826	0.628	0.593	0.605	0.590	0.861	0.851	0.824	0.801	0.745	0.745	
gemma3-12	FOLIO	0.313	0.326	0.352	0.317	0.283	0.701	0.721	0.745	0.706	0.711	0.777	0.667	0.643	0.653	0.623	0.986	0.984	0.978	0.988	0.994	0.735	0.745	0.775	0.773	0.772	0.918	0.911	0.900	0.892	0.887	
	LogicalDeduction	0.313	0.326	0.326	0.287	0.257	0.701	0.706	0.706	0.750	0.711	0.777	0.653	0.667	0.723	0.700	0.986	0.938	0.942	0.938	0.928	0.737	0.732	0.757	0.747	0.745	0.919	0.920	0.900	0.891	0.898	
	ProntoQA	0.313	0.300	0.296	0.309	0.317	0.701	0.691	0.716	0.647	0.711	0.757	0.577	0.623	0.593	0.640	0.986	0.910	0.968	0.956	0.878	0.737	0.648	0.605	0.633	0.572	0.918	0.912	0.908	0.913	0.903	
	ProofWriter	0.313	0.326	0.274	0.304	0.304	0.706	0.760	0.740	0.789	0.755	0.773	0.653	0.670	0.637	0.630	0.986	0.992	0.996	0.996	0.988	0.738	0.757	0.755	0.767	0.773	0.920	0.910	0.901	0.894	0.889	
	gsm8k	0.296	0.339	0.304	0.274	0.322	0.706	0.711	0.696	0.721	0.701	0.773	0.777	0.620	0.690	0.640	0.986	0.978	0.982	0.982	0.988	0.738	0.732	0.753	0.745	0.738	0.920	0.868	0.631	0.565	0.544	
gemma3-27	FOLIO	0.330	0.365	0.391	0.387	0.326	0.721	0.765	0.779	0.721	0.804	0.827	0.867	0.857	0.843	0.867	0.984	0.992	0.986	0.988	0.990	0.763	0.798	0.802	0.782	0.813	0.936	0.937	0.932	0.936	0.930	
	LogicalDeduction	0.335	0.417	0.383	0.400	0.370	0.725	0.716	0.711	0.716	0.745	0.833	0.793	0.790	0.840	0.837	0.984	0.988	0.992	0.982	0.767	0.777	0.768	0.777	0.768	0.936	0.933	0.934	0.930	0.927	0.927	
	ProntoQA	0.330	0.383	0.352	0.348	0.387	0.725	0.750	0.740	0.716	0.730	0.830	0.840	0.827	0.840	0.830	0.986	0.992	0.988	0.992	0.994	0.763	0.773	0.740	0.748	0.752	0.936	0.922	0.929	0.931	0.925	
	ProofWriter	0.348	0.361	0.370	0.404	0.378	0.730	0.760	0.760	0.760	0.765	0.833	0.870	0.843	0.863	0.853	0.986	1.000	0.994	0.994	0.990	0.770	0.785	0.802	0.790	0.787	0.936	0.929	0.936	0.926	0.92	

Model	Source	AR-LSAT					FOLIO					LogicalDeduction					ProntoQA					ProofWriter					gsm8k					
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	
qwen3-4	FOLIO	0.726	0.665	0.674	0.678	0.678	0.804	0.819	0.809	0.828	0.809	0.967	0.953	0.943	0.947	0.947	0.998	0.998	1.000	0.998	0.998	0.998	0.817	0.810	0.805	0.802	0.822	0.917	0.914	0.902	0.906	0.901
	LogicalDeduction	0.696	0.730	0.683	0.661	0.678	0.804	0.784	0.775	0.784	0.967	0.980	0.960	0.950	0.967	0.980	1.000	1.000	1.000	1.000	1.000	0.810	0.813	0.805	0.914	0.908	0.905	0.916	0.910	0.910		
	ProntoQA	0.735	0.717	0.648	0.665	0.670	0.804	0.789	0.809	0.804	0.833	0.967	0.933	0.943	0.953	0.943	0.998	0.990	0.998	0.992	0.998	0.817	0.793	0.798	0.793	0.787	0.917	0.901	0.906	0.909	0.907	
	ProofWriter	0.735	0.717	0.670	0.674	0.678	0.804	0.750	0.760	0.775	0.779	0.967	0.940	0.957	0.967	0.957	0.998	0.998	0.996	0.998	0.996	0.817	0.802	0.820	0.823	0.800	0.917	—	0.921	0.910	0.909	
qwen3-8	FOLIO	0.683	—	—	—	—	—	—	—	—	—	—	—	—	—	1.000	0.996	1.000	1.000	1.000	—	—	—	—	—	0.932	0.922	0.920	0.919	0.916		
	LogicalDeduction	0.683	0.691	0.683	0.713	0.730	0.779	0.789	0.809	0.799	0.799	0.950	0.980	0.990	0.977	0.973	1.000	1.000	0.998	0.998	0.872	0.860	0.818	0.840	0.857	0.932	0.920	0.920	0.919	0.923		
	ProntoQA	0.683	0.674	0.704	0.704	0.717	0.779	0.828	0.814	0.819	0.848	0.950	0.887	0.890	0.860	0.907	1.000	0.982	0.992	0.998	0.996	0.872	0.838	0.835	0.825	0.823	0.932	0.924	0.932	0.920	0.918	
	gsm8k	0.683	0.722	0.713	0.739	0.717	0.779	0.804	0.789	0.799	0.789	0.950	0.943	0.930	0.947	0.973	1.000	0.998	0.996	0.996	0.994	0.872	0.855	0.885	0.887	0.932	0.930	0.936	0.931	0.939		
qwen3-14	FOLIO	0.791	0.757	0.726	0.743	0.730	0.828	0.828	0.833	0.838	0.993	0.990	0.983	0.973	0.990	0.982	0.996	1.000	0.998	0.996	0.838	0.820	0.815	0.820	0.820	0.876	0.912	0.916	0.910	0.910	0.910	
	LogicalDeduction	0.791	0.783	0.774	0.743	0.739	0.828	0.838	0.848	0.833	0.983	0.983	0.987	0.983	0.987	0.982	0.986	0.982	0.982	0.982	0.970	0.838	0.802	0.798	0.815	0.807	0.876	0.914	0.912	0.913	0.907	
	ProntoQA	0.791	0.748	0.761	0.761	0.722	0.828	0.833	0.848	0.824	0.828	0.993	0.990	0.973	0.993	0.990	0.982	0.998	1.000	0.998	0.838	0.798	0.807	0.822	0.808	0.876	0.918	0.914	0.908	0.920	0.911	
	ProofWriter	0.791	0.704	0.722	0.700	0.726	0.828	0.838	0.833	0.819	0.838	0.993	—	—	—	0.982	1.000	1.000	0.994	0.998	0.838	0.837	0.847	0.840	0.835	0.876	0.913	0.910	0.907	0.911	0.911	
qwen3-32	FOLIO	0.791	0.770	0.752	0.800	0.770	0.828	0.828	0.814	0.824	0.794	0.993	0.997	0.987	0.987	0.982	0.976	0.980	0.974	0.964	0.964	0.838	0.833	0.823	0.843	—	0.876	0.920	0.927	0.926	0.923	
	LogicalDeduction	0.752	0.739	0.748	0.757	0.752	—	—	—	—	0.913	0.600	0.637	0.683	0.707	1.000	1.000	1.000	1.000	1.000	0.860	0.863	0.863	0.857	0.845	0.929	0.932	0.933	0.932	0.939		
	ProntoQA	0.787	0.739	0.717	0.730	0.674	0.804	0.770	0.819	0.824	0.819	0.890	0.610	0.527	0.490	0.520	—	—	—	0.828	0.852	0.857	0.843	0.852	0.843	0.852	0.934	0.932	0.930	0.929		
	gsm8k	0.774	0.735	0.730	0.700	0.713	0.804	0.770	0.833	0.799	0.824	0.910	0.690	0.580	0.623	0.573	1.000	0.998	1.000	1.000	0.996	—	—	—	—	0.932	0.936	0.930	0.931	—	—	
llama3-8	FOLIO	0.261	0.226	0.209	0.217	0.226	0.603	0.627	0.588	0.608	0.618	0.527	0.507	0.480	0.440	0.467	0.782	0.848	0.782	0.766	0.794	0.457	0.520	0.523	0.520	0.523	0.614	0.575	0.575	0.546	0.541	
	LogicalDeduction	0.261	0.257	0.213	0.204	0.235	0.603	0.549	0.569	0.593	0.574	0.527	0.510	0.500	0.530	0.480	0.784	0.834	0.792	0.846	0.838	0.457	0.463	0.513	0.525	0.620	0.604	0.591	0.576	0.570		
	ProntoQA	—	—	—	—	—	—	—	—	—	0.527	0.500	0.450	0.457	0.493	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	gsm8k	0.261	0.248	0.274	0.252	0.230	0.603	0.583	0.593	0.564	0.583	0.527	0.470	0.503	0.463	0.782	0.844	0.826	0.838	0.810	0.457	0.562	0.593	0.592	0.583	0.614	0.587	0.575	0.569	0.548		
llama3-8	FOLIO	0.261	0.317	0.222	0.239	0.200	0.603	0.583	0.549	0.495	0.578	0.527	0.567	0.507	0.470	0.450	0.782	0.772	0.776	0.782	0.770	0.457	0.512	0.518	0.478	0.480	0.614	0.663	0.714	0.732	0.714	
	LogicalDeduction	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	ProntoQA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	gsm8k	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

Table 11: RAG_bm25 cross-domain EM of Qwen models. Red numbers indicate improvements over 0-shot.

Model	Source	AR-LSAT					FOLIO					LogicalDeduction					ProntoQA					ProofWriter					gsm8k				
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4
qwen3	FOLIO	0.230	0.209	0.222	0.226	0.230	—	—	—	—	—	0.453	0.517	0.467	0.463	0.470	0.864	0.864	0.866	0.822	0.846	0.533	0.573	0.555	0.535	0.543	0.745	0.713	0.704	0.685	0.668
	LogicalDeduction	0.226	0.183	0.178	0.209	0.226	0.623	0.667	0.637	0.618	0.593	—	—	—	—	—	0.854	0.856	0.856	0.842	0.828	0.510	0.528	0.557	0.517	0.542	0.738	0.701	0.702	0.675	0.669
	ProntoQA	0.274	0.170	0.209	0.174	0.187	0.450	0.475	0.446	0.456	0.430	0.450	0.420	0.433	0.430	—	—	—	—	—	—	0.497	0.418	0.360	0.312	0.292	0.737	0.698	0.693	0.717	0.694
	ProofWriter	0.230	0.217	0.183	0.209	0.213	0.623	0.603	0.676	0.632	0.642	0.447	0.513	0.467	0.467	0.473	0.862	0.850	0.892	0.868	0.854	—	—	—	—	0.738	0.717	0.712	0.691	0.667	
	gsm8k	0.274	0.213	0.239	0.178	0.217	0.647	0.588	0.588	0.588	0.608	0.447	0.457	0.453	0.450	0.447	0.862	0.758	0.766	0.660	0.638	0.512	0.468	0.448	0.402	0.422	—	—	—	—	
qwen7	FOLIO	0.217	0.248	0.239	0.283	0.257	—	—	—	—	—	0.637	0.617	0.683	0.617	0.630	0.978	0.962	0.980	0.982	0.974	0.632	0.625	0.650	0.672	0.632	0.884	0.892	0.893	0.884	
	LogicalDeduction	0.248	0.243	0.278	0.278	0.265	0.711	0.632	0.706	0.701	0.681	—	—	—	—	—	0.978	0.972	0.982	0.974	0.978	0.633	0.613	0.612	0.622	0.630	0.884	0.890	0.889	0.883	
	ProntoQA	0.209	0.248	0.226	0.248	0.252	0.662	0.618	0.667	0.632	0.642	0.637	0.550	0.653	0.627	0.620	—	—	—	—	—	0.627	0.613	0.627	0.603	0.650	0.887	0.863	0.878	0.883	
	ProofWriter	0.222	0.248	0.283	0.248	0.239	0.701	0.711	0.696	0.701	0.706	0.653	0.630	0.650	0.650	0.637	0.976	0.970	0.970	0.984	0.986	—	—	—	—	—	0.889	0.871	0.873	0.862	
	gsm8k	0.243	0.274	0.257	0.283	0.265	0.662	0.574	0.647	0.642	0.652	0.643	0.603	0.633	0.607	0.597	0.932	0.924	0.924	0.924	0.932	0.627	0.627	0.625	0.622	0.605	—	—	—	—	
qwen14	FOLIO	0.343	0.348	0.274	0.335	0.304	0.681	0.750	0.745	0.765	0.760	0.820	0.793	0.780	0.820	0.773	0.942	0.962	0.968	0.968	0.972	0.688	0.715	0.735	0.738	0.732	0.938	0.931	0.932	0.936	
	LogicalDeduction	0.330	0.348	0.317	0.365	0.343	0.696	0.686	0.740	0.706	0.725	0.797	0.810	0.807	0.810	0.807	0.956	0.964	0.976	0.968	0.974	0.680	0.702	0.698	0.700	0.707	0.936	0.930	0.934	0.931	
	ProntoQA	0.322	0.304	0.313	0.339	0.296	0.721	0.745	0.755	0.701	0.696	0.787	0.767	0.767	0.763	0.743	0.948	0.972	0.978	0.982	0.984	0.678	0.622	0.637	0.607	0.577	0.933	0.929	0.926	0.918	
	ProofWriter	0.304	0.335	0.270	0.343	0.370	0.672	0.735	0.750	0.755	0.745	0.813	0.787	0.787	0.807	0.803	—	—	—	—	—	0.974	0.677	0.738	0.752	0.762	0.936	0.937	0.927	0.931	
	gsm8k	0.322	0.304	0.357	0.274	0.291	0.711	0.686	0.706	0.706	0.686	0.810	0.780	0.817	0.783	0.773	0.948	0.940	0.900	0.894	0.880	0.688	0.620	0.607	0.605	0.583	0.922	0.924	0.923	0.926	
qwen32	FOLIO	0.348	0.374	0.339	0.357	0.357	—	—	—	—	—	0.850	0.847	0.837	0.880	0.890	1.000	0.998	0.998	1.000	0.998	0.718	0.693	0.742	0.748	0.735	0.893	0.909	0.906	0.895	
	LogicalDeduction	0.348	0.357	0.309	0.343	0.330	0.725	0.745	0.750	0.721	0.750	—	—	—	—	—	1.000	0.996	0.996	0.998	0.998	0.688	0.733	0.728	0.725	0.738	0.892	0.906	0.912	0.906	
	ProntoQA	0.304	0.365	0.322	0.322	0.322	0.261	0.745	0.745	0.701	0.672	0.686	0.857	0.830	0.813	0.817	0.823	—	—	—	—	0.695	0.675	0.682	0.680	0.682	0.892	0.886	0.905	0.911	
	ProofWriter	0.335	0.313	0.322	0.352	0.348	0.750	0.725	0.755	0.735	0.730	0.847	0.873	0.860	0.873	0.877	—	—	—	—	—	—	—	—	—	—	0.889	0.898	0.912	0.880	
	gsm8k	0.326	0.317	0.335	0.339	0.322	0.740	0.716	0.711	0.696	0.716	0.840	0.870	0.837	0.807	0.783	0.998	0.996	0.994	0.988	0.988	0.703	0.613	0.617	0.618	0.628	—	—	—	—	
gemma3-4	FOLIO	0.274	0.300	0.243	0.226	0.243	—	—	—	—	—	0.627	0.623	0.600	0.627	0.603	0.918	0.930	0.930	0.944	0.912	0.650	0.618	0.643	0.635	0.642	—	—	—	—	
	LogicalDeduction	0.235	0.261	0.283	0.235	0.248	0.618	0.618	0.613	0.603	0.588	—	—	—	—	—	0.920	0.876	0.878	0.856	0.892	0.642	0.532	0.562	0.532	0.547	—	—	—	—	
	ProntoQA	0.243	0.230	0.230	0.261	0.248	0.623	0.608	0.564	0.598	0.603	0.600	0.623	0.613	0.587	0.633	—	—	—	—	—	0.647	0.620	0.625	0.575	0.455	0.863	0.850	0.839	0.844	
	ProofWriter	0.287	0.270	0.265	0.243	0.261	0.608	0.642	0.637	0.652	0.618	0.610	0.587	0.600	0.623	0.597	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	gsm8k	0.248	0.235	0.230	0.248	0.265	0.637	0.593	0.623	0.598	0.588	0.593	0.647	0.567	0.580	0.583	0.912	0.868	0.846	0.834	0.818	0.647	0.568	0.573	0.578	0.553	—	—	—	—	
gemma3-12	FOLIO	0.304	0.287	0.291	0.322	0.330	—	—	—	—	—	0.757	0.753	0.767	0.770	0.763	0.992	0.980	0.982	0.988	0.984	0.740	0.762	0.768	0.753	0.778	—	—	—	—	
	LogicalDeduction	0.317	0.274	0.304	0.270	0.322	0.725	0.706	0.740	0.725	0.686	—	—	—	—	—	0.986	0.984	0.990	0.986	0.982	0.747	0.738	0.755	0.753	0.768	—	—	—	—	
	ProntoQA	0.313	0.361	0.339	0.287	0.287	0.706	0.701	0.706	0.681	0.716	0.767	0.740	0.757	0.747	0.720	—	—	—	—	—	0.750	0.650	0.622	0.652	0.597	0.917	0.909	0.916	0.901	
	ProofWriter	0.317	0.274	0.309	0.287	0.335	0.735	0.745	0.779	0.775	0.755	0.773	0.757	0.747	0.773	0.800	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	gsm8k	0.296	0.278	0.335	0.300	0.335	0.730	0.652	0.676	0.701	0.667	0.763	0.747	0.773	0.777	0.793	0.986	0.970	0.974	0.988	0.996	0.752	0.698	0.717	0.737	0.747	—	—	—	—	
gemma3-27	FOLIO	—	0.322	—	0.409	—	—	—	—	—	—	0.823	0.830	0.857	0.890	0.857	0.986	0.988	0.988	0.996	0.994	0.767	0.810	0.820	0.812	0.808	—	—	—	—	
	LogicalDeduction	0.374	0.400	0.370	0.391	0.387	0.706	0.760	0.735	0.755	0.735	—	—	—	—	—	0.984	0.984	0.990	0.986	0.992	0.760	0.772	0.792	0.770	0.767	—	—	—	—	
	ProntoQA	0.357	0.378	0.370	0.361	0.352	0.721	0.735	0.711	0.696	0.735	0.820	0.867	0.870	0.863	0.863	—	—	—	—	—	0.765	0.757	0.735	0.737	0.718	0.935	0.927	0.934	0.932	
	ProofWriter	0.383	0.400	0.387	0.378	0.352	0.716	0.765	0.784	0.789	0.765	0.823	0.847	0.853	0.850	0.877	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
	gsm8k	0.370	0.361	0.383	0.339	0.374	0.721	0.716	0.711	0.716	0.716	0.830	0.903	0.890	0.883	0.860	0.984	0.982	—	—	—	0.980	0.978	0.773	0.735	0.732	0.742	—	—	—	

Table 12: ConE 1

Model	Source	AR-LSAT				FOLIO				LogicalDeduction				ProntoQA				ProofWriter				gsm8k										
		0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	0	1	2	3	4	
qwen3-4	FOLIO	0.722	0.678	0.657	0.674	0.700	—	—	—	—	—	0.970	0.940	0.960	0.923	0.920	1.000	0.994	0.992	0.996	0.990	0.823	0.823	0.823	0.812	0.908	0.909	0.897	0.904	0.914		
	LogicalDeduction	0.726	0.700	0.704	0.709	0.713	0.804	0.804	0.775	0.784	0.819	—	—	—	—	—	1.000	0.996	1.000	1.000	0.998	0.803	0.798	0.805	0.823	0.800	0.916	0.912	0.916	0.915		
	ProntoQA	0.726	0.643	0.691	0.696	0.678	0.779	0.765	0.799	0.775	0.770	0.967	0.930	0.933	0.927	0.907	—	—	—	—	—	0.822	0.785	0.782	0.767	0.757	0.917	0.906	0.905	0.876		
	ProofWriter	0.722	0.635	0.665	0.670	0.665	0.804	0.819	0.819	0.824	0.804	0.967	0.967	0.927	0.940	0.940	1.000	0.996	0.996	0.994	0.994	—	—	—	—	—	0.917	0.902	0.885	0.882	0.893	
	gsm8k	0.713	0.678	0.652	0.691	0.687	0.779	0.740	0.770	0.775	0.750	0.960	0.967	0.953	0.947	0.957	1.000	0.996	0.998	0.998	0.998	0.820	0.793	0.800	0.807	0.818	—	—	—	—	—	
qwen3-8	FOLIO	0.630	0.722	0.691	0.691	0.665	—	—	—	—	—	0.937	0.977	0.960	0.970	0.970	1.000	1.000	1.000	1.000	1.000	0.878	0.872	0.860	0.838	0.860	0.928	0.926	0.922	0.917	0.917	
	LogicalDeduction	0.630	0.717	0.713	0.713	0.665	0.779	0.814	0.828	0.824	0.809	—	—	—	—	—	1.000	0.996	1.000	1.000	0.998	0.878	0.855	0.853	0.855	0.850	0.931	0.928	0.920	0.926	0.923	
	ProntoQA	0.691	0.713	0.730	0.678	0.674	0.814	0.814	0.799	0.789	0.828	0.943	0.957	0.933	0.903	0.910	—	—	—	—	—	0.875	0.867	0.858	0.830	0.850	0.926	0.925	0.923	0.918	0.922	
	ProofWriter	0.630	0.687	0.691	0.687	0.691	0.784	0.824	0.809	0.799	0.814	0.927	0.970	0.967	0.963	0.963	1.000	0.998	1.000	1.000	1.000	—	—	—	—	—	—	0.926	0.922	0.925	0.921	0.917
	gsm8k	0.635	0.730	0.700	0.735	0.722	0.814	0.799	0.794	0.789	0.814	0.923	0.953	0.967	0.973	0.970	1.000	0.996	0.998	0.998	0.998	0.872	0.878	0.870	0.872	0.875	—	—	—	—	—	

Table 13: ConE 2

Setting	Model Reasoning and Output
Context	A Japanese game company created the game The Legend of Zelda. All games on the Top 10 list are made by Japanese game companies. If a game sells more than one million copies, then it will be included in the Top 10 list. The Legend of Zelda sold more than one million copies.
Zero-shot	<i>Since the Legend of Zelda sold more than one million copies, it should be included in the Top 10 list. However, it is stated that the Legend of Zelda is not on the Top 10 list. This creates a contradiction with the information that all games on the Top 10 list are made by Japanese game companies and the Legend of Zelda is made by a Japanese game company. Therefore, the statement "The Legend of Zelda is not on the Top 10 list" must be true. Final answer: A</i> Final answer: A (True)
Demostration	Question: In the last 25 years, 60% of Scrabble champions have been women and the rest have been men. There is one champion per year. Of the men, 40% had a beard. How many men with beards have been Scrabble champion? Reasoning: 40% of champions are men because $100 - 60 = 40$ 40 champions have been men because $25 \times .4 = 10$ 10 champions have been men with beards because $10 \times .4 = 4$
Cross-domain	<i>Since the Legend of Zelda sold more than one million copies and all games on the Top 10 list are made by Japanese game companies, and the Legend of Zelda is made by a Japanese game company, it must be on the Top 10 list. Therefore, the statement "The Legend of Zelda is not on the Top 10 list" is false. Final answer: B</i> Final answer: B (False)

Table 14: Comparison between zero-shot reasoning and cross-domain reasoning on a cross-domain example. DIN-guided retrieval corrects the logical inconsistency made by zero-shot prompting.

Setting	Model Reasoning and Output
Context	<p>People in this club who perform in school talent shows often attend and are very engaged with school events. People in this club either perform in school talent shows often or are inactive and disinterested community members. People in this club who chaperone high school dances are not students who attend the school. All people in this club who are inactive and disinterested members of their community chaperone high school dances. All young children and teenagers in this club who wish to further their academic careers and educational opportunities are students who attend the school. Bonnie is in this club and she either both attends and is very engaged with school events and is a student who attends the school or is not someone who both attends and is very engaged with school events and is not a student who attends the school.</p>
Zero-shot	<p><i>Let's break down the information step by step: 1. If Bonnie chaperones high school dances, then she is not a student who attends the school (from the third statement). 2. If Bonnie does not chaperone high school dances, then she performs in school talent shows often (from the second statement). 3. If Bonnie performs in school talent shows often, then she is very engaged with school events (from the first statement). 4. If Bonnie is very engaged with school events, then she is a student who attends the school (from the first statement). 5. From steps 1 and 4, we can conclude that Bonnie cannot be both a student who attends the school and chaperone high school dances. 6. From steps 2 and 3, we can conclude that if Bonnie does not chaperone high school dances, then she is a student who attends the school. 7. From steps 5 and 6, we can conclude that Bonnie is not a student who attends the school if she chaperones high school dances, and she is a student who attends the school if she does not chaperone high school dances. 8. From the fourth statement, we can conclude that if Bonnie is a student who attends the school, then she is a young child or teenager who wishes to further her academic career and educational opportunities. 9. From steps 7 and 8, we can conclude that Bonnie is a young child or teenager who wishes to further her academic career and educational opportunities if she does not chaperone high school dances. 10. From the fifth statement, we can conclude that if Bonnie is an inactive and disinterested member of the community, then she chaperones high school dances. 11. From steps 7 and 10, we can conclude that Bonnie is an inactive and disinterested member of the community if she chaperones high school dances. 12. From steps 9 and 11, we can conclude that Bonnie is both a young child or teenager who wishes to further her academic career and educational opportunities and an inactive and disinterested member of the community if she does not chaperone high school dances. Therefore, the statement is true. Final answer: A</i> Final answer: A (True)</p>
Demonstration	<p>Question: Forty percent of the students have elected to learn from home during the pandemic. The remaining students are divided into two equal groups, only one of which is physically in school on any day. What percent of students are present in school? Reasoning: $100 - 40 = 60$ percent of students are remaining. $60 / 2 = 30$ percent are present in school.</p>
Cross-domain	<p><i>Let's break down the information step by step: 1. If Bonnie chaperones high school dances, then she is not a student who attends the school (from the context). 2. If Bonnie does not chaperone high school dances, then she performs in school talent shows often (from the given statement). 3. If Bonnie performs in school talent shows often, then she is very engaged with school events (from the context). 4. If Bonnie is very engaged with school events, then she is a student who attends the school (from the context). 5. From steps 1 and 4, we have a contradiction: Bonnie cannot both be a student who attends the school and not be a student who attends the school. 6. Therefore, the given statement "If Bonnie either chaperones high school dances or, if she does not, she performs in school talent shows often" must be false, because it leads to a contradiction. Final answer: B</i> Final answer: B (False)</p>

Table 15: Comparison between zero-shot reasoning and cross-domain reasoning on a cross-domain example. DIN-guided retrieval corrects the logical inconsistency made by zero-shot prompting.

Setting	Model Reasoning and Output
Context	<p>The bear does not chase the lion. The bear visits the lion. The bear visits the tiger. The lion chases the bear. The lion chases the tiger. The lion visits the bear. The rabbit chases the tiger. The rabbit is big. The rabbit is not green. The rabbit is young. The rabbit visits the tiger. The tiger chases the rabbit. The tiger is rough. The tiger visits the bear. If something chases the bear and it sees the rabbit then the bear does not visit the rabbit. If something chases the lion then the lion is rough. If the rabbit chases the tiger and the tiger visits the rabbit then the rabbit is not big. If something chases the lion then the lion does not visit the tiger. If something chases the rabbit and it chases the tiger then the rabbit chases the bear. If something is young and it does not see the rabbit then it sees the bear. If something visits the lion then it is young. If the tiger visits the bear and the bear is big then the tiger visits the lion. Young things are big.</p>
Question	<p>Based on the above information, is the following statement true, false, or unknown? The bear visits the lion.</p>
Zero-shot	<p><i>Let's break down the information step by step: 1. The bear needs the cow. 2. The cow needs the rabbit. 3. The cow sees the bear. 4. The cow visits the lion. 5. The lion is nice. 6. The lion is red. 7. The lion does not see the cow. 8. The lion visits the cow. 9. The rabbit sees the cow. 10. The rabbit visits the cow. 11. If something needs the cow and the cow visits the rabbit then it visits the cow. 12. If the lion needs the bear then the bear sees the cow. 13. If the lion sees the bear and the bear is rough then the bear does not see the cow. 14. If something visits the lion then the lion visits the bear. 15. If something visits the bear then it needs the bear. 16. If something visits the bear then the bear needs the cow. 17. If something sees the cow then it visits the bear. From the information, we know: - The rabbit sees the cow and visits the cow. - According to point 11, if the lion needs the bear, then the bear sees the cow. - According to point 13, if the lion sees the bear and the bear is rough, then the bear does not see the cow. - Since the lion visits the cow (point 8), and the rabbit visits the cow (point 9), the rabbit needs the cow (point 15). - According to point 16, if the rabbit visits the bear, then the bear needs the cow. - Since the bear needs the cow (point 1), and the rabbit needs the cow (point 15), the bear needs the cow. - According to point 13, if the lion sees the bear and the bear is rough, then the bear does not see the cow. But we know the bear sees the cow (point 3). - Therefore, the bear is not rough (point 13). So, the rabbit is not rough.</i></p> <p>Final answer: A (True)</p>
Demonstration	<p>Context: No reptile has fur. All snakes are reptiles. Question: Based on the above information, is the following statement true, false, or uncertain? Some snake has fur. Options: A) True B) False C) Uncertain Reasoning: 1. "No reptile has fur" means: If something is a reptile, it does not have fur. 2. "All snakes are reptiles" means: If something is a snake, it is a reptile. 3. If all snakes are reptiles, and no reptile has fur, then no snake has fur. 4. The statement "Some snake has fur" would be false, because there is no snake that has fur. The correct option is: B Answer: B</p>

Table 16: FOLIO-ProofWriter

Setting	Model Reasoning and Output
Cross-domain	<p>1. The bear needs the cow. 2. The cow needs the rabbit. 3. The cow sees the bear. 4. The cow visits the lion. 5. The lion is nice. 6. The lion is red. 7. The lion does not see the cow. 8. The lion visits the cow. 9. The rabbit sees the cow. 10. The rabbit visits the cow. 11. If something needs the cow and the cow visits the rabbit, then it visits the cow. 12. If the lion needs the bear, then the bear sees the cow. 13. If the lion sees the bear and the bear is rough, then the bear does not see the cow. 14. If something visits the lion, then the lion visits the bear. 15. If something visits the bear, then it needs the bear. 16. If something visits the bear, then the bear needs the cow. 17. If something sees the cow, then it visits the bear. From the given information, we know: - The rabbit sees the cow (point 9). - The rabbit visits the cow (point 10). According to point 11, if the rabbit visits the cow, then the rabbit visits the cow. This is a tautology and doesn't provide new information. According to point 12, if the lion needs the bear, then the bear sees the cow. However, the lion does not see the cow (point 7), so the lion cannot need the bear. Therefore, the bear does not see the cow. According to point 13, if the lion sees the bear and the bear is rough, then the bear does not see the cow. Since the lion does not see the cow, this condition does not apply. According to point 14, if the rabbit visits the lion, then the lion visits the bear. However, the rabbit does not visit the lion (point 10), so this condition does not apply. According to point 15, if the rabbit visits the bear, then the rabbit needs the bear. But the rabbit does not visit the bear (point 10), so this condition does not apply. According to point 16, if the rabbit visits the bear, then the bear needs the cow. But the rabbit does not visit the bear (point 10), so this condition does not apply. According to point 17, if the rabbit sees the cow, then the rabbit visits the bear. But the rabbit does not see the cow (point 9), so this condition does not apply. Since none of the conditions point to the rabbit being rough, and we have no information that directly or indirectly states the rabbit is rough, the statement "The rabbit is rough" is unknown.</p> <p>Final answer: C</p>

Table 17: FOLIO-ProofWriter

Setting	Model Reasoning and Output
Context	Bob is big. Bob is red. Bob is smart. Erin is quiet. Fiona is big. Fiona is blue. Fiona is cold. Fiona is smart. Gary is red. Gary is smart. If Bob is quiet then Bob is cold. If someone is red then they are quiet. If someone is red and quiet then they are cold. All kind people are big. If someone is red and blue then they are kind. All cold, quiet people are blue. Cold, big people are kind. If Erin is big then Erin is cold. If someone is cold and blue then they are smart.
Question	Based on the above information, is the following statement true, false, or unknown? Gary is red.
Zero-shot	<i>Let's analyze the information step by step: 1. Context Analysis: - Bob is quiet. - Charlie is blue, cold, and nice. - Dave is blue and cold. - Erin is blue. - If something is blue and cold, then it is big. - Round, nice things are big. - All nice, quiet things are furry. - If Bob is round, then Bob is furry. - If Bob is furry and Bob is nice, then Bob is big. - If something is big and blue, then it is quiet. - If something is blue, then it is nice. - All nice things are cold. 2. Determine Erin's Characteristics: - Erin is blue. - From the context, if something is blue, then it is nice. - Therefore, Erin is nice. 3. Check if Erin is Quiet: - The context does not explicitly state that Erin is quiet or not. - There is no direct information linking Erin to being quiet or not being quiet. 4. Conclusion: - Since there is no information provided about Erin being quiet or not, we cannot determine if the statement "Erin is quiet" is true or false. Final answer: B</i>
Demostration	Tumpuses are bright. Tumpuses are yumpuses. Yumpuses are sweet. Yumpuses are wumpuses. Wumpuses are not transparent. Wumpuses are vumpuses. Each vumpus is angry. Vumpuses are rompuses. Rompuses are happy. Every rompus is a zumpus. Zumpuses are brown. Zumpuses are numpuses. Numpuses are not large. Impuses are not happy. Numpuses are jompuses. Jompuses are earthy. Jompuses are dumpuses. Polly is a tumpus. Question: Is the following statement true or false? Polly is not happy. Options: A) True B) False Reasoning: Polly is a tumpus. Tumpuses are yumpuses. Polly is a yumpus. Yumpuses are wumpuses. Polly is a wumpus. Wumpuses are vumpuses. Polly is a vumpus. Vumpuses are rompuses. Polly is a rompus. Rompuses are happy. Polly is happy. Answer: B Context: Impuses are not fruity. Impuses are wumpuses. Each wumpus is not temperate. Wumpuses are dumpuses. Rompuses are not sweet. Dumpuses are kind. Dumpuses are zumpuses. Zumpuses are wooden. Zumpuses are vumpuses. Every vumpus is large. Vumpuses are yumpuses. Yumpuses are transparent. Yumpuses are numpuses. Numpuses are brown. Numpuses are tumpuses. Tumpuses are sweet. Tumpuses are jompuses. Wren is a zumpus. Question: Is the following statement true or false? Wren is sweet. Options: A) True B) False Reasoning: Wren is a zumpus. Zumpuses are vumpuses. Wren is a vumpus. Vumpuses are yumpuses. Wren is a yumpus. Yumpuses are numpuses. Wren is a numpus. Numpuses are tumpuses. Wren is a tumpus. Tumpuses are sweet. Wren is sweet. Answer: A
Cross-domain	<i>Let's analyze the information step by step: 1. Erin is blue. 2. If something is blue, then it is nice. → Erin is nice. 3. Erin is cold (not explicitly stated, but we can infer this from the context since blue and cold are often associated). 4. Round, nice things are big. → Since we don't know if Erin is round, we can't conclude Erin is big. 5. All nice things are cold. → This is consistent with Erin being cold. 6. All nice things are cold. → This is consistent with Erin being cold. 7. If something is blue and cold, then it is big. → Since Erin is blue and cold, Erin is big. 8. If something is big and blue, then it is quiet. → Since Erin is big and blue, Erin is quiet. The statement "Erin is quiet" is true based on the given information. Final answer: A</i>

Table 18: ProntoQA-ProofWriter