Big Reasoning with Small Models: Instruction Retrieval at Inference Time

Anonymous ACL submission

Abstract

Small language models (SLMs) enable low-cost, private, on-device inference, but they often fail on problems that require specialized domain knowledge or multi-step reasoning. Existing approaches for improving SLM reasoning either rely on scale (e.g., chain-of-thought prompting), require task-specific training that limits reuse and generality (e.g., distillation), or retrieve unstructured information that still leaves the SLM to determine an appropriate reasoning strategy. We propose instruction retrieval, an inference-time intervention that augments an SLM with structured, reusable reasoning procedures rather than raw passages. We construct an Instruction Corpus by clustering similar training questions and using a teacher model to generate generalizable guides that pair domain background with explicit step-by-step procedures. At inference, the SLM retrieves the instructions most relevant to a given query and executes the associated procedures without any additional fine-tuning. Across three challenging domains—medicine, law, and mathematics, instruction retrieval yields consistent gains for models with at least 3B parameters, improving accuracy by 9.4%, 7.9%, and 5.1%, respectively, with the strongest 14B model surpassing GPT-4o's zero-shot performance on knowledge-intensive tasks.

1 Introduction

Large language models (LLMs) have demonstrated remarkable generalization and reasoning ability across a wide range of domains, from mathematical problem solving (Cobbe et al., 2021) and clinical diagnostics (Kwon et al., 2024) to legal and commonsense reasoning (Hendrycks et al., 2021a). Much of their success stems from massive parameter scale and diverse pretraining data, which allow them to internalize both factual knowledge and multi-step reasoning procedures. However, scaling to hundreds of billions of parameters introduces substantial costs. For example, serving a single 175B-parameter model requires over 300GB of GPU memory and specialized infrastructure (Frantar et al., 2023; Zheng et al., 2022), making real-time or on-prem deployment difficult. Moreover, production-grade LLMs typically operate as closed-source APIs, raising privacy and governance concerns in sensitive domains such as medicine and law. Small language models offer a practical alternative. SLMs can be deployed locally, audited directly, and fine-tuned for specific domains (Pham et al., 2024; Belcak et al., 2025; Fu et al., 2023). Their compact size lowers cost and latency while enabling operation on commodity hardware, including laptops and edge devices. Moreover, recent work (Srivastava et al., 2025) challenges the notion that reasoning is exclusive to large-scale models and that smaller models can achieve competitive reasoning performance when trained or compressed effectively. Yet, their reasoning behavior remains underexplored (Zhu et al., 2024), and unlike LLMs, which encode broad world knowledge and reasoning patterns within their parameters, SLMs still have a limited capacity to internalize such information (Fu et al., 2023). This gap motivates the central question of this work: how can small, efficient models achieve strong reasoning performance without scaling up their parameters or relying on additional training for every new task?

We address this question through instruction intervention at inference time, which augments small models with explicit, retrievable reasoning procedures. Instead of generating reasoning steps it cannot reliably produce on its own, the model retrieves structured instructions generated by a larger model that pairs domain background with step-by-step guidance. As illustrated in Figure 1, this approach externalizes reasoning as a retrieval process, supplying the scaffolds that SLMs lack while retaining their efficiency and privacy advantages. The method is entirely inference-based: a single, unmodified SLM retrieves and follows instructions without additional training, and the approach generalizes across domains.

We evaluate this framework across three reasoning benchmarks (MedQA, MMLU Law, and MathQA) to assess both its effectiveness and the conditions under which it provides the greatest benefit. Specifically, we investigate two questions. First, does instruction retrieval reliably improve the reasoning performance of small language models? Second, how does this benefit depend on model capacity and instruction design (e.g., length)? We show that across all three benchmarks, instruction retrieval improves accuracy by 5–10 percentage points for models with at least 3B parameters, with-

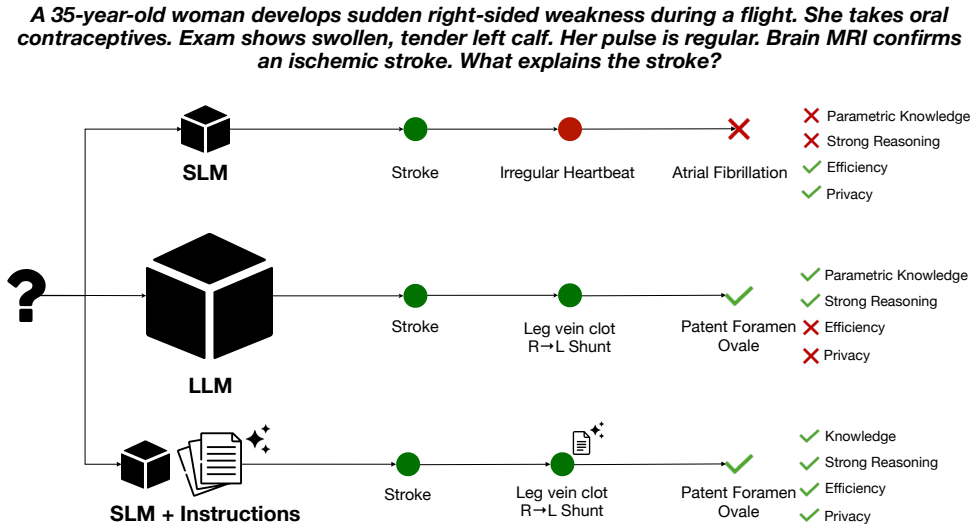

Figure 1: Comparison of reasoning steps across model configurations on a MedQA-style case. The small language model alone produces an incorrect chain, while a large model reaches the right diagnosis but at high computational and privacy cost. Adding retrieved instructions supplies the missing background and procedural steps, enabling the SLM to reproduce the correct reasoning efficiently and privately.

out any additional fine-tuning Notably, on knowledge-intensive tasks such as MedQA and MMLU Law, a 14B-parameter model equipped with retrieved instructions surpasses GPT-4o in zero-shot accuracy. Further analysis shows that concise insteuctions yield the largest gains while performance remains stable even when instructions are shared across broader categories of problems. This robustness indicates that the approach does not depend on finely tailored instructions for each instance. To isolate the sources of improvement, we conduct a mixed-effects analysis and find that gains depend more on a model's ability to follow structured guidance than on parameter count alone. In several cases, a smaller but more instruction-capable model outperforms a larger model with weaker underlying performance. These results suggest that externalizing reasoning as retrievable text offers a practical path to scaling reliable inference on resource-limited or privacy-sensitive hardware. All code and the Instruction Corpus will be released upon acceptance to support reproducibility.

2 Related Work

Chain of Thought. Chain-of-Thought (CoT) prompting encourages models to decompose a reasoning task into a sequence of intermediate steps rather than attempting to produce an answer directly. This approach has been shown to substantially improve the performance of LLMs across commonsense, symbolic, and mathematical reasoning benchmarks (Wei et al., 2023; Kojima et al., 2023; Wang et al., 2023). These gains, however, rely heavily on scale: only models with tens or hundreds of billions of parameters such as PaLM 540B (Chowdhery et al., 2022) or GPT-3 175B (Brown et al., 2020) , reliably benefit from CoT prompting. Smaller models frequently generate illogical reasoning traces, and their accuracy can even decline when forced to produce step-by-step rationales (Wei et al., 2023).

Distillation. To overcome the scale dependence of prompting, a parallel line of work distills reasoning traces from LLMs into SLMs. These methods fine-tune smaller models on CoT-style rationales generated by larger models or derived from labeled data, with the goal of internalizing step-by-step reasoning skills (Ho et al., 2023; Li et al., 2022; Magister et al., 2023; Fu et al., 2023; Hsieh et al., 2023). Distillation has proven effective on arithmetic and symbolic reasoning benchmarks such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b), where reasoning patterns can be learned from repeated structures. However, the outcomes are less promising for knowledge-intensive tasks where accurate rationales depend on factual knowledge that smaller models struggle to retain. Because SLMs have limited parameter capacity, attempts to train them with extensive domain knowledge often lead to overspecialization and reduced generality (Fu et al., 2023).

Retrieval. Rather than encoding all knowledge directly into model parameters, retrieval-based methods expand a model's capacity by supplementing it with external information at inference time. Classical dense retrieval (Karpukhin et al., 2020) improves open-domain question answering by retrieving relevant passages, but the returned evidence is often noisy or unstructured, leaving the model to extract and organize reasoning steps on its own. Smaller models are more easily overloaded by long or noisy contexts and are less capable of abstracting structure from raw passages. As a result, the effectiveness of retrieval for SLMs depends on ensuring that the retrieved evidence is relevant and struc-

Variant	Avg. Length	Know. Comp.	Know. Rel.	Sample Snippet (MedQA)
High School Concise	453	4.64	4.83	*. . . See the big clue (underweight athlete with missed periods), match it to the rule (FHA → low estrogen), then pick "low bone density". . .*
Graduate Concise	631	4.95	4.84	*. . . Prioritize the diagnostic triad (amenorrhea + low BMI + training), apply HPO suppression logic, reject violated conditions (e.g., high TSH/estrogen), and select "decreased bone density". . .*
High School Verbose	1,408	4.99	4.76	*. . . Restate the task, list key clues (negative hCG, low BMI, training, lanugo), link them to FHA (low GnRH → LH/FSH↓ → estrogen↓), explain why distractors fail, and conclude with "decreased bone density". . .*
Graduate Verbose	1,765	5.00	4.53	*. . . Synthesize the amenorrhea–energy-deficit pattern, map it via leptin/kisspeptin → GnRH suppression to hypoestrogenism, weigh alternative axes (thyroid, prolactin, PCOS) by contradiction, and justify "decreased bone density" as the compelled outcome. . .*

Table 1: Comparison of four instruction variants differing in audience level and length. Columns report the average token length and Claude quality scores (five-point Likert) for knowledge comprehensiveness and relevance, with example excerpts from MedQA. Concise variants are 2–3× shorter while retaining high relevance, whereas verbose variants expand coverage at the cost of slightly lower relevance.

tured. More recent generate-then-retrieve frameworks address this limitation by synthesizing targeted context before retrieval (Yu et al., 2023; Wang et al., 2025), yielding more relevant evidence for reasoning. While these approaches have been developed for LLMs, their lessons are even more pertinent for SLMs.

Knowledge Augmentation. Recent work (Kang et al., 2023; Zhao et al., 2024) improves on unstructured retrieval by more tightly coupling retrieved evidence with the reasoning process itself. Knowledge-Augmented Reasoning Distillation (Kang et al., 2023) grounds LLM-style rationales in retrieved passages and uses a reranker to emphasize evidence that directly supports those rationales. This helps smaller models learn reasoning that is explicitly tied to external knowledge, but it still relies on task-specific fine-tuning and encodes the resulting reasoning policy in model parameters. Probe then Retrieve and Reason (Zhao et al., 2024) modularizes this pipeline by separating it into two distilled SLMs: a probing model that identifies what knowledge is needed and formulates retrieval queries, and a reasoning model that constructs step-by-step rationales from the retrieved passages. While this separation improves interpretability, the reasoning procedure remains implicit in the model's learned behavior and must be retrained to adapt to new domains.

3 Instruction Corpus

Small language models often fail on complex inference because they lack both sufficient domain knowledge and the structured support needed for multi-step reasoning. To address this gap, we build an *Instruction Corpus*, a collection of modular instructions for each domain. Each instruction has two parts: (i) background knowledge relevant to a problem type and (ii) step-by-step reasoning procedures for solving it. At inference, instructions are retrieved and included in the prompt, guiding reasoning without task-specific fine-tuning. For example, in MedQA, a group of training questions may ask about contraindications for anticoagulants. From this group, we derive an instruction that outlines how to check mechanisms of action, review contraindications, and compare relative risks.

Corpus Construction We construct the Instruction Corpus in three stages: clustering, instruction generation, and retrieval. First, training examples from each benchmark are embedded with OpenAI's `text-embedding-3-large` and grouped using agglomerative clustering with average linkage and cosine distance. The resulting dendrogram supports different levels of granularity, where lower thresholds yield broad categories and higher thresholds produce highly specific clusters; by default, we adopt the most fine-grained partition so that each cluster corresponds to a distinct reasoning skill. Next, for each cluster, we generate a reusable instruction by prompting GPT-5 with standardized templates (D) and up to 5 question examples from the cluster. The input examples serve only to guide generation and do not appear verbatim in the final instructions. Finally, at inference time, test queries are embedded with the same encoder and matched by cosine similarity to the closest clusters; the top-k instructions (default $k = 5$) are retrieved and included in prompt to provide both factual grounding and procedural scaffolding for the given problem type. See Appendix 2 for threshold values and statistics.

Instruction Variants While instructions provide missing knowledge and reasoning scaffolds, their effectiveness depends on how they are written. We therefore study how variation in audience level and length shapes small-model performance. For the audience level, we create two variants. A graduate-level ver-

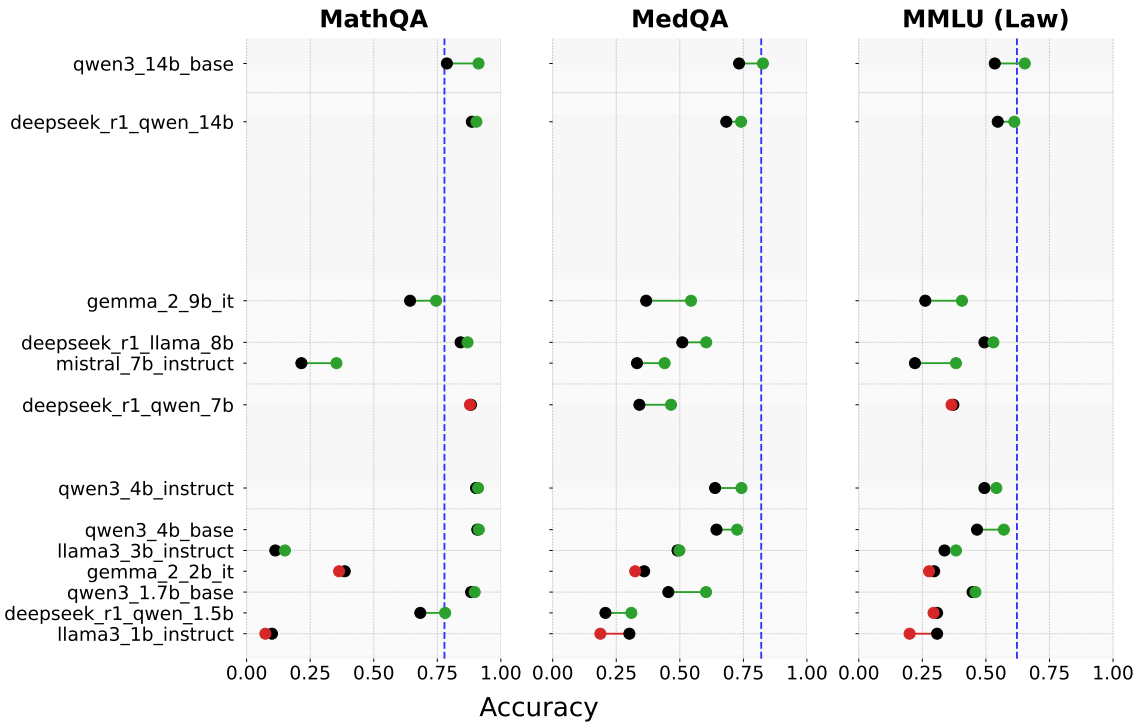

Figure 2: Accuracy of SLMs with and without instruction retrieval across three reasoning benchmarks. Each pair shows zero-shot accuracy (black) and performance with High-School Concise instructions (green = improvement, red = decline). The dashed blue line marks GPT-4o's zero-shot accuracy. Instruction retrieval yields consistent gains once models exceed 3B parameters, especially on MedQA and MMLU (Law), where knowledge and procedural reasoning are most demanding. Appendix Table 5 reports the same results in tabular form for reference.

sion preserves domain terminology and assumes advanced conceptual familiarity (e.g., clinical reasoning in MedQA). The high-school-level version simplifies vocabulary and logical complexity (*i.e., explain this to me like a high schooler*). Aligning readability with prior knowledge reduces intrinsic cognitive load and improves comprehension, a core prediction of Cognitive Load Theory (Renkl & Atkinson, 2003; Sweller, 2024). Along the depth axis, we contrast concise instructions that contain only the essential reasoning steps with longer instructions that include additional commentary to stimulate germane load and schema formation. Moreover, recent prompt-engineering studies show that prompt compression often preserves, and sometimes even enhances, downstream accuracy despite large token reductions, which underscores the practical value of concise variants (Renze & Guven, 2024; Li et al., 2024). Crossing the two factors yields four prompt variants per task. Descriptive quality summaries by style are reported in Appendix Table 4. We also include a baseline prompt that does not enforce any constraints on style or length. This version reflects the instructions produced directly from our generation pipeline and serves as a neutral reference point. Comparing the four controlled variants against this baseline helps us to isolate the effects of readability and verbosity from artifacts of the generation process. Full prompt templates for all conditions, including the baseline, are provided in Appendix D.

Quality Validation Not all instructions are equally useful. Generic prompts (e.g., 'think carefully') lack a stable, inspectable problem-solving procedure. By contrast, the documents in our corpus provide explicit domain background and concrete reasoning steps that define a consistent and inspectable decision process. Here, we verify that instructions are high quality and that the four variants differ only in style and length. To do so, we evaluate each instruction on three axes: knowledge (coverage and relevance of background facts), reasoning (soundness and task-specificity of steps), and clarity (structured, unambiguous presentation). A detailed rubric is given in Appendix A.1.3. Each document is scored on a five-point Likert scale by Claude Sonnet 4.5 to avoid same-model bias. We repeat evaluations three times and report averaged scores; full results are shown in Table 4. Across tasks, mean scores are consistently high (4.6–5.0) on all three axes, confirming that the corpus provides accurate and reliable scaffolds rather than noisy or generic text. Table 1 summarizes quality differences across variants: verbosity increases comprehensiveness but slightly reduces relevance, while audience level has only minor effects. Clarity remains uniformly high across all conditions. Thus, the variants are equivalent in structure and quality, differing only in the controlled stylistic factors of length and audience.

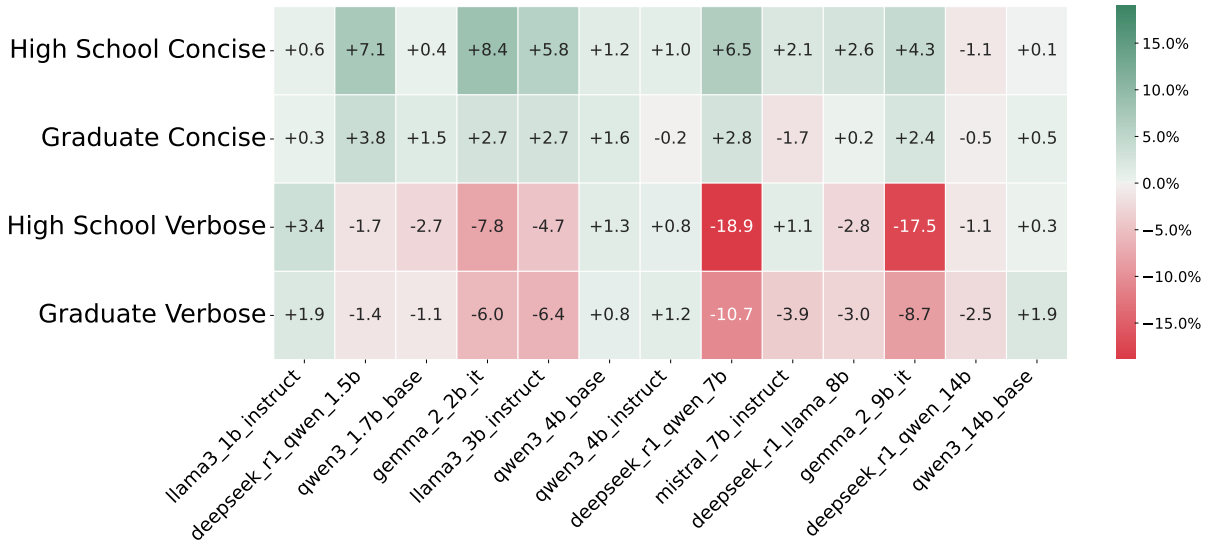

Figure 3: Accuracy differences when enforcing specific instruction styles compared to baseline instructions with no style or length constraints. Concise instructions generally outperform the baseline, while verbose instructions tend to reduce accuracy. Audience level (High School vs. Graduate) has smaller and less consistent effects.

Corpus Profile The Instruction Corpus varies systematically in both length and size across tasks. Concise instructions average 500–750 tokens, while verbose instructions expand to 1.3k–2.1k tokens, a two- to three-fold increase in prompt length. Length distributions are shown in Appendix Figure 6, with per-task breakdowns in Figure 7. Graduate-level instructions are slightly longer and more detailed than high-school versions, but this effect is small relative to the impact of verbosity. Corpus size is determined by the clustering granularity used during construction. At the default, most fine-grained threshold, the corpus contains approximately 29k instruction groups for MathQA, 8.4k for MedQA, and 1.0k for MMLU Law. Group sizes are highly skewed: most clusters contain only a single example, while the largest include up to 12. As a result, many instructions are narrowly scoped and grounded in one or a few examples, capturing the specific cues and reasoning steps needed for that problem type. Increasing the clustering threshold merges these singletons into larger groups, producing more general instructions that apply across broader classes of questions. We return to this tradeoff in Section 6, where we show that instruction retrieval remains robust across a wide range of cluster sizes. Full statistics and length distributions are reported in Appendix 2 and Appendix A.1.1.

4 Experimental Setup

We evaluate the effect of the Instruction corpus on three benchmarks and publicly available models spanning five major open families, and compare performance against GPT-4o as a reference LLM.

4.1 Tasks

We use three benchmarks that collectively stress different dimensions of reasoning. MedQA (Jin et al., 2020) contains multiple-choice questions from professional medical board exams; success requires both factual recall of biomedical knowledge and diagnostic reasoning. MMLU Professional Law (Hendrycks et al., 2021a) focuses on legal exam questions, testing recall of statutes and precedents alongside structured case-based reasoning. MathQA (Amini et al., 2019) evaluates symbolic reasoning through math word problems, requiring models to translate natural language into computational procedures. For MedQA, we additionally compare against a standard RAG baseline using the benchmark's provided medical textbooks. The other two tasks do not allow for a comparable RAG baseline without constructing a new external corpus. Given a test question, the model retrieves the top-5 relevant passages using the same embedding model (`text-embedding-3-large`) and cosine similarity; the retrieved passages are provided in the prompt at inference time.

4.2 Models

We evaluate across thirteen models from five major open families: Llama 3 (AI@Meta, 2024), Gemma 2 (Team, 2024), Qwen 3 (Team, 2025), Mistral (Jiang et al., 2023), and DeepSeek R1 Distilled (DeepSeek-AI, 2025). Model sizes span from 1B to 14B parameters, providing coverage across the typical range of deployable SLMs. For Llama 3, Gemma 2, and Mistral we use publicly released instruction-tuned variants. For DeepSeek R1 we evaluate distilled models released in sizes from 1.5B to 14B. For Qwen 3, we evaluate three model sizes (1.7B, 4B, and 14B), using the default

thinking-enabled configuration. At inference, each test question is embedded using the same model used during clustering (`text-embedding-3-large`) and matched to training clusters by cosine similarity. The top five cluster-level instructions are retrieved and inserted into the prompt before decoding. To ensure comparability, all prompts and inference settings were standardized across tasks, with generation performed at a temperature of 0.7 and a top-p value of 0.95. GPT-4o, accessed via API, is included as a reference point to situate results against LLM performance.

5 Results

Instruction retrieval reliably improves the reasoning ability of small language models across domains and architectures. Gains emerge once models reach a minimum capacity of around 3B parameters and grow with scale, ranging from 5 to 18 percentage points over zero-shot prompting. Improvements are largest on knowledge-intensive tasks such as MedQA and MMLU Law. To understand what drives these gains, we next examine how instruction design, model family, and prompt length shape performance, beginning with aggregate accuracy across tasks.

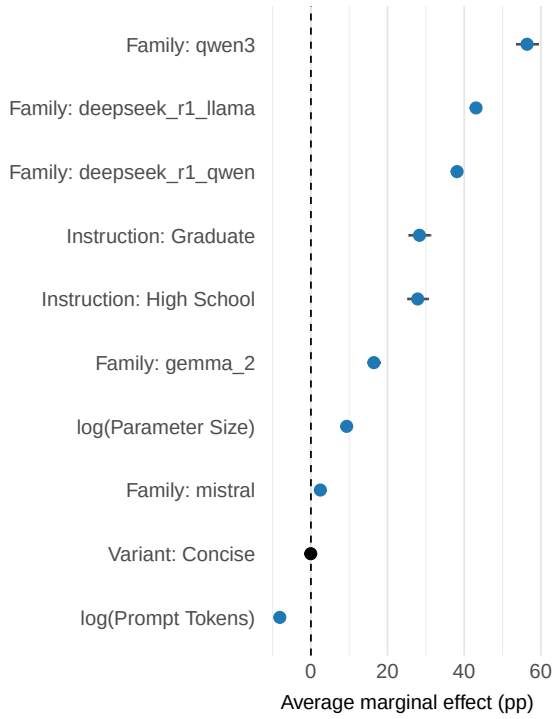

Figure 4: Marginal effects on accuracy relative to zero-shot prompting. All instruction variants yield large gains (+28–29pp). Family effects dominate: Qwen3 (+56pp) and DeepSeek R1 (+38–43pp) show the strongest ability to follow instructions compared to the LLaMA-3 reference, while Gemma-2 (+16pp) and Mistral (+2pp) are smaller. Model scale adds a consistent +9pp per log-unit increase in parameters, while longer prompts reduce accuracy by –8pp per log-token.

5.1 Aggregate Performance

Figure 2 shows the accuracy of the High School Concise instructions compared to zero-shot prompting across MathQA, MedQA, and MMLU Law. For the smallest models under 3B parameters, improvements are mixed and often negative. A few exceptions exist, such as Qwen-3 1.7B base, which gains +15pp on MedQA, but the same model shows negligible changes on MMLU and MathQA. This instability suggests that tiny models lack the intrinsic reasoning ability to reliably leverage retrieved scaffolds, which is in line with prior research (Li et al., 2025). At around 3B parameters, gains become consistently positive, even if modest. LLaMA-3 3B, for instance, improves by +1–5pp across benchmarks. Once a model is large enough to follow multi-step guidance, retrieved instructions shift from being noise to providing usable structure and information. Notably, both instruction retrieval and passage-based RAG reduce accuracy for models below 3B parameters. Once models cross this threshold, instruction retrieval yields substantially larger gains than RAG on MedQA (+9.3pp vs. +3.2pp on average), suggesting that structured procedural guidance is more effective than unstructured evidence for small-model reasoning. Above this threshold, performance consistently improves for instructions. Models from 4B to 14B shows gains across all three tasks, with deltas typically in the +5–18pp range. The largest improvements appear in MedQA and MMLU Law, where domain-specific knowledge and reasoning may lack representation in the limited parameter space. For example, Gemma-2 9B improves by +14pp on MMLU Law and Mistral 7B by +16pp, while MedQA gains reach +18pp at 9B. In contrast, MathQA gains are more muted, though still positive, likely because mathematical reasoning is already well represented in pretraining data, leaving less headroom for improvement. Overall, retrieved instructions consistently boost accuracy once models exceed 3B parameters, with robust gains across both symbolic and domain-expert reasoning tasks. Importantly, on MedQA and MMLU Law, the 14B parameter SLMs with retrieved instructions surpass the zero-shot accuracy of GPT-4o.

5.2 Effects of Instruction Variation

Here, we compare the four instruction variations against a baseline condition in which models receive retrieved instructions without any explicit style or length constraints to learn more about the effects of how instructions are written, rather than the content of the instructions themselves. Figure 3 shows accuracy deltas across all variants. A consistent pattern is that concise instructions outperform the baseline, while verbose instructions often reduce accuracy, particularly for larger models. This finding mirrors our instruction-quality analysis: longer prompts increase comprehensiveness but dilute relevance, creating unnecessary cognitive load for models that are already capable of multi-

step reasoning.

Audience effects are smaller and less consistent. In some cases, simplified high school variants provide a slight benefit, especially on MedQA, where complex biomedical terminology may overwhelm smaller models. In other cases, graduate-level detail performs equally well or marginally better. Overall, these results suggest that the primary determinant of instruction effectiveness is length rather than audience level. Concise scaffolds reduce context overhead while still supplying the essential reasoning steps, aligning with our quality evaluation that verbosity may introduce tangential material. The negligible effect of audience framing implies that once instructions are well structured, SLMs can adapt to register differences without measurable loss. Indicating that instruction retrieval benefits from minimizing extraneous detail rather than tailoring explanations to a presumed audience.

5.3 Determinants of Instruction Effectiveness

To isolate which properties of instructions and models drive performance gain we fit a mixed-effects logistic regression at the question–model–variant level (Figure 4). The dependent variable is a binary indicator of whether the model answered a test question correctly and fixed effects include instruction variant (audience and length), prompt length, model size, and a zero-shot indicator. Interactions with model size test whether stylistic effects vary with capacity. Random intercepts for questions and datasets account for variation in item difficulty and domain. LLaMA-3 serves as the reference family, chosen because it is the oldest model in our pool. Coefficients are reported as average marginal effects, which represent the change in predicted accuracy when a predictor increases by one unit while other variables are held constant. Positive values indicate improvements relative to the reference categories.

Both High School and Graduate instruction variants yield large and significant gains of +28–29pp over zero-shot prompting, confirming that retrieved scaffolds substantially improve small-model reasoning. Yet among fixed effects, model family dominates: Qwen-3 shows the largest positive offset (+56pp), followed by DeepSeek R1–LLaMA (+43pp) and DeepSeek R1–Qwen (+38pp). These gains are substantially larger than the +9pp associated with a log-unit increase in size, highlighting that architecture and pretraining choices matter as much as, or more than, scale. Families like Qwen and DeepSeek R1 may already have stronger step-following and reasoning performance out of the box compared to the older Llama3 variants. Even with the same parameter count, families differ in attention implementations, training recipes, or RLHF alignment, all of which can influence how reliably they follow structured prompts. Prompt length has a negative effect. Each log-unit increase in tokens reduces accuracy by –8pp, highlighting that verbosity imposes a systematic penalty. While conciseness does not improve accuracy on average, its interaction with size is positive: larger models benefit disproportionately from brevity, while smaller models gain little. Verbose instructions add no measurable value, again suggesting that detail beyond essential steps creates redundancy rather than a usable signal.

6 Ablation and Analysis

While instruction retrieval consistently improves small-model reasoning, its effectiveness depends on design choices such as corpus granularity and construction cost. We analyze these factors on MedQA using the High School Concise variant and models of at least 3B parameters, focusing on robustness to clustering choices and the amortized cost of corpus construction.

6.1 Cluster Size

The instruction corpus groups training questions into clusters using agglomerative clustering with cosine distance thresholds (Appendix Table 2). Each clustering threshold determines how close examples must be to form a cluster, with lower values producing many small, fine-grained clusters and higher values merging examples into broader, more general groups. From a design perspective, smaller clusters may yield highly specific instructions that fit narrow problem types but limit reuse, while larger clusters may promote generalization at the risk of losing task-specific cues. We therefore evaluate how this granularity influences downstream accuracy. Figure 5 reports average MedQA accuracy across models as a function of mean cluster size. Accuracy remains stable as clusters become broader, indicating that instructions can represent larger groups of questions without loss in performance. This robustness suggests that clustering granularity can be treated as a tunable hyperparameter: by adjusting the threshold on a development set, we can identify the coarsest grouping that preserves accuracy. Optimizing this allows the use of fewer instructions that are more representative of topics rather than individual questions and may improve interpretability. Per-model accuracy across thresholds is reported in Appendix Table 6.

6.2 Amortized Cost Analysis

Instruction retrieval introduces a one-time processing cost to construct the Instruction Corpus, after which inference proceeds with a fixed retrieval overhead. This design shifts computation from repeated generation to amortized preparation, making the approach cost-effective even at moderate corpus sizes. We estimate corpus construction costs using GPT-5.1, assuming an average of 300 input tokens and 600 output tokens per instruction, reflecting our empirical finding that concise instructions perform best. At current pricing ($1.25 per million input tokens and $10.00 per million output tokens), generating a single instruction costs approximately $0.0064. Under these assumptions, a corpus of 1,000 instructions costs $6.38 to generate, while

a corpus of 10,000 instructions costs $63.75. Embedding costs are negligible in comparison. Using text-embedding-3-large at $0.065 per million tokens (batched), embedding the entire corpus adds less than $1, even at the largest corpus sizes. Importantly, our ablation results show that coarser clustering, and thus fewer, more general instructions, perform comparably to highly bespoke corpora. In practice, corpora in the 1k–3k range achieve most of the observed gains, placing total one-time costs well below $20. Importantly, this cost is incurred only once and can be amortized across an unlimited number of users and inference runs. For example, a hospital system could construct an instruction corpus derived from clinical guidelines for diagnosing heart failure, which could then be reused indefinitely by cardiologists across multiple hospitals without additional generation cost.

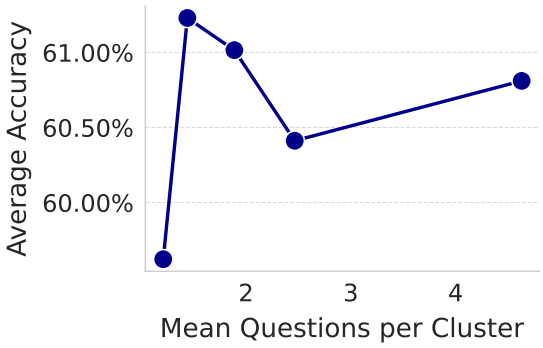

Figure 5: Effect of clustering granularity on MedQA accuracy, averaged across all evaluated models. Performance remains consistent across thresholds, indicating that instruction retrieval is robust even when clusters merge into broader groups. Broken down by model size in Appendix 8

7 Discussion

Our results show that small language models above a minimal capacity threshold can reliably incorporate retrieved instructions to perform multi-step reasoning, substantially narrowing the gap to large models without additional training. This finding aligns with prior evidence (Li et al., 2025) that models below roughly 3B parameters struggle to internalize long reasoning traces, but can benefit from shorter, externally provided guidance. Once models cross this threshold, retrieved instructions shift from being noise to providing usable structure, suggesting that instruction retrieval exploits reasoning capabilities that are present but underutilized in small models. Although instruction retrieval yields consistent gains, the mechanisms underlying these improvements remain an open question. One plausible explanation is that retrieved instructions supply highly targeted domain knowledge that small models cannot reliably recall from their parameters alone. Another is that the explicit procedural structure reduces search during generation by constraining the order and priority of reasoning steps, leading to more stable and coherent inference. In practice, these effects are likely complementary: instructions both surface relevant information and impose a reasoning policy that small models can execute more reliably than unconstrained chain-of-thought generation.

Our results also suggest that instruction retrieval should not be applied indiscriminately. Gains are largest on knowledge-intensive domains such as medicine and law, where both factual recall and structured decision-making exceed the capacity of small models. In contrast, improvements on mathematics are smaller, reflecting limited headroom in domains where models already exhibit strong internal reasoning. These findings motivate several directions for future work. In the current framework, instructions are retrieved for every query, even when a model may already be capable of solving the problem unaided. A natural extension is competence-aware retrieval, in which models estimate their confidence or uncertainty and invoke instructions only when external guidance is likely to improve reasoning. More broadly, because instructions are external, non-parametric artifacts, instruction corpora need not be static. Future systems could support self-evolving instruction collections that update over time as models encounter new failure modes, domain knowledge changes, or revised best practices, enabling continuous improvement without retraining.

8 Conclusion

This work reframes reasoning as a retrieval problem. Rather than requiring small language models to internally generate or store specialized domain knowledge or complex reasoning chains, we show that they can retrieve and execute structured instructions that combine domain knowledge with explicit procedures. Across MedQA, MMLU Law, and MathQA, instruction retrieval consistently improves multi-step reasoning for models above a minimal capacity threshold, without any additional training. Concise instructions yield the largest gains, and effectiveness depends more on a model's ability to follow structured guidance than on parameter count alone. Unlike chain-of-thought prompting, which relies on scale, or distillation, which requires task-specific training and often reduces generality, instruction retrieval externalizes reasoning as reusable text. Compared to standard retrieval-augmented generation, which provides unstructured information, instructions define an explicit and inspectable problem-solving strategy. By decoupling reasoning knowledge from model parameters, instruction retrieval enables small models to approach large-model performance while preserving the efficiency and privacy of local inference. A single instruction corpus can be reused and updated independently of the model, offering a practical and maintainable path toward reliable reasoning with compact models.

Limitations

Several limitations remain. The current corpus is derived from benchmark datasets rather than the open-ended problems encountered in real-world use. Extending this framework to domain-scale or continuously evolving environments raises a broader question: how should instructions and their associated knowledge be represented, retrieved, and maintained in the wild rather than within fixed benchmarks? Our retrieval quality also depends on a simple top-k similarity search; future work could incorporate re-ranking or adaptive selection to improve relevance and coverage as the corpus grows.

References

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms, 2019. URL https://arxiv.org/abs/1905.13319.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small language models are the future of agentic ai, 2025. URL https://arxiv.org/abs/2506.02153.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022. URL https://arxiv.org/abs/2204.02311.

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

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Yao Fu, Hao Peng, Litu Ou, Ashish Sabharwal, and Tushar Khot. Specializing smaller language models towards multi-step reasoning, 2023. URL https://arxiv.org/abs/2301.12726.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021b. URL https://arxiv.org/abs/2103.03874.

Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers, 2023. URL https://arxiv.org/abs/2212.10071.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. URL https://arxiv.org/abs/2305.02301.

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

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams, 2020. URL https://arxiv.org/abs/2009.13081.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks, 2023. URL https://arxiv.org/abs/2305.18395.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916.

Taeyoon Kwon, Kai Tzu iunn Ong, Dongjin Kang, Seungjun Moon, Jeong Ryong Lee, Dosik Hwang, Yongsik Sim, Beomseok Sohn, Dongha Lee, and Jinyoung Yeo. Large language models are clinical reasoners: Reasoning-aware diagnosis framework with prompt-generated rationales, 2024. URL https://arxiv.org/abs/2312.07399.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. Explanations from large language models make small reasoners better, 2022. URL https://arxiv.org/abs/2210.06726.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners, 2025. URL https://arxiv.org/abs/2502.12143.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey, 2024. URL https://arxiv.org/abs/2410.12388.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason, 2023. URL https://arxiv.org/abs/2212.08410.

Thang M. Pham, Phat T. Nguyen, Seunghyun Yoon, Viet Dac Lai, Franck Dernoncourt, and Trung Bui. Slimlm: An efficient small language model for on-device document assistance, 2024. URL https://arxiv.org/abs/2411.09944.

Alexander Renkl and Robert K. Atkinson. Structuring the transition from example study to problem solving in cognitive skill acquisition: A cognitive load perspective. *Educational Psychologist*, 38(1): 15–22, 2003. doi: 10.1207/S15326985EP3801_3.

Matthew Renze and Erhan Guven. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pp. 476–483. IEEE, November 2024. doi: 10.1109/fllm63129.2024.10852493. URL http://dx.doi.org/10.1109/FLLM63129.2024.10852493.

Gaurav Srivastava, Shuxiang Cao, and Xuan Wang. Towards reasoning ability of small language models, 2025. URL https://arxiv.org/abs/2502.11569.

John Sweller. Cognitive load theory and individual differences. *Learning and Individual Differences*, 102: 102423, 2024. doi: 10.1016/j.lindif.2024.102423. Open access.

Gemma Team. Gemma. 2024. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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

Yubo Wang, Xueguang Ma, and Wenhu Chen. Augmenting black-box llms with medical textbooks for biomedical question answering, 2025. URL https://arxiv.org/abs/2309.02233.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators, 2023. URL https://arxiv.org/abs/2209.10063.

Yichun Zhao, Shuheng Zhou, and Huijia Zhu. Probe then retrieve and reason: Distilling probing and reasoning capabilities into smaller language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 13026–13032, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.1140/.

Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica. Alpa: Automating inter- and Intra-Operator parallelism for distributed deep learning. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pp. 559–578, Carlsbad, CA, July 2022. USENIX Association. ISBN 978-1-939133-28-1. URL https://www.usenix.org/conference/osdi22/presentation/zheng-lianmin.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models, 2024. URL https://arxiv.org/abs/2308.07633.

A Appendix

A.1 Instruction Corpus

A.1.1 Length Distributions

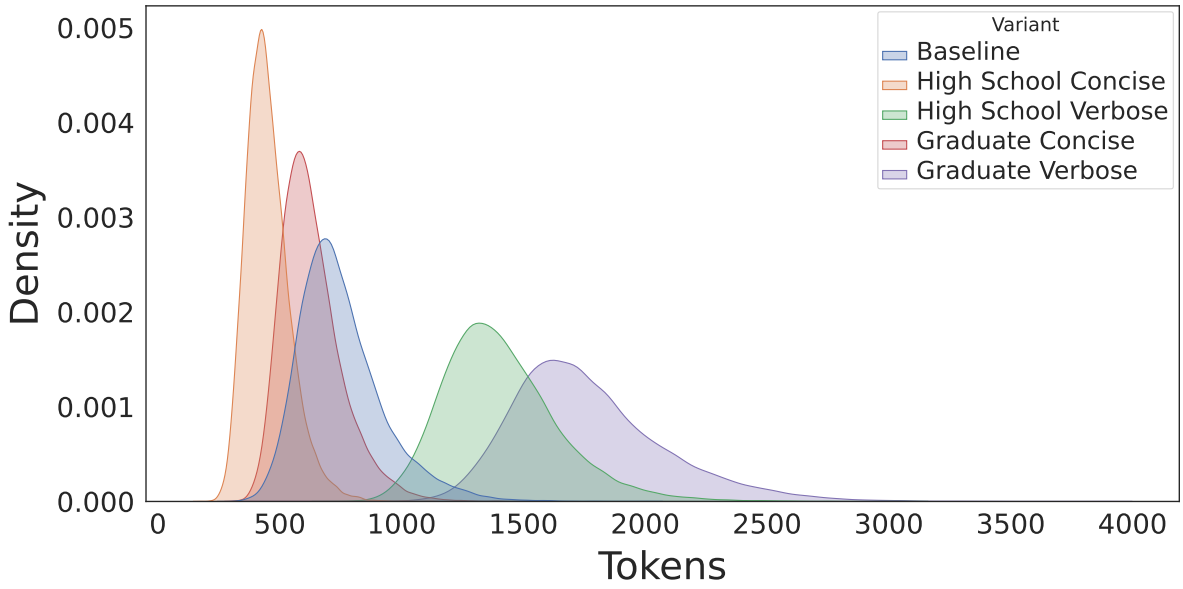

Figure 6: Distribution of instruction lengths across all variants. Concise instructions average 500–750 tokens, while verbose versions expand to 1.3k–2.1k tokens, a consistent two- to threefold increase

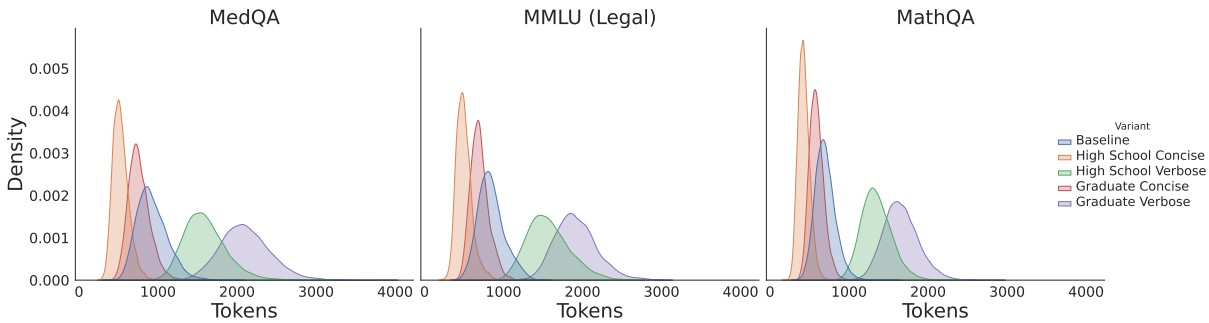

Figure 7: Distribution of instruction lengths across all variants. and broken down by task.

A.1.2 Clustering Details

Task	Threshold	# Clusters	Mean Size	Std. Size	Max Size	Silhouette
MedQA	0.177	8414	1.21	0.63	11	0.079
	0.209	7070	1.44	1.08	18	0.107
	0.244	5390	1.89	1.83	20	0.127
	0.272	4130	2.46	2.74	38	0.135
	0.325	2198	4.63	6.34	62	0.128
MMLU	0.246	1048	1.30	0.79	12	0.083
	0.275	872	1.57	1.29	21	0.101
	0.322	586	2.33	2.67	36	0.101
	0.352	410	3.33	4.45	49	0.099
	0.635	3	455.7	639.5	1360	0.123
MathQA	0.008	29093	1.03	0.22	7	0.024
	0.063	15949	1.87	1.88	30	0.387
	0.093	12229	2.44	2.81	61	0.480
	0.154	8509	3.51	4.97	140	0.539
	0.255	4789	6.23	6.23	786	0.442

Table 2: Cluster statistics across thresholds for MedQA, MMLU (Law), and MathQA. We report the number of clusters, mean and standard deviation of cluster size, maximum cluster size, and silhouette score (higher = more cohesive clusters).

A.1.3 Evaluation Criteria

Criterion	Score	Descriptor
Knowledge Comprehensiveness	5	All background facts needed for typical instances are present, including key definitions, edge cases, and disambiguations.
	4	Nearly all essentials covered. Minor omissions that rarely affect correctness.
	3	Mixed coverage with several common cases or definitions missing. Sometimes blocks a correct solution.
	2	Multiple essential facts missing. Frequent failure without extra knowledge.
	1	Largely incomplete background.
Knowledge Relevance	5	Background contains only necessary or highly useful facts for the cluster. No tangents.
	4	Small amount of extra detail that is not distracting.
	3	Noticeable extraneous content. Can distract or slow reasoning.
	2	Large amount of irrelevant or low-yield content. Likely to mislead.
	1	Mostly off-topic or generic background.
Reasoning Accuracy	5	Reasoning steps are logically sound, factually correct, and properly sequenced. No contradictions.
	4	Minor imprecision or wording issues that do not change the outcome.
	3	At least one underspecified or brittle step that could lead to a wrong branch. Small gaps.
	2	Clear logical or factual error that would often yield an incorrect answer.
	1	Reasoning is largely incorrect or inconsistent.
Reasoning Relevance	5	Steps are tailored to the problem type, align with input and output, and include cluster-critical operations.
	4	Mostly tailored with minimal generic filler.
	3	Mix of tailored and generic steps. Some do not map to the task structure.
	2	Largely generic advice. Missing one or more cluster-critical steps.
	1	Steps unrelated to the problem type.
Clarity	5	Concise, well structured, unambiguous. Consistent terminology. Numbered steps or clear bullets. Explicit stop conditions or decision points.
	4	Generally clear with minor verbosity or mild ambiguity.
	3	Mixed clarity. Some steps vague or terminology inconsistent.
	2	Hard to follow. Long sentences, unclear step boundaries, or undefined terms.
	1	Confusing or unreadable.

Table 3: Five-point Likert rubric for instruction quality. Global decision rules: cap Reasoning Accuracy at 2 if any factual error is present in the steps. Cap Reasoning Relevance at 2 and Knowledge Comprehensiveness at 3 if a required step is missing. Cap Knowledge Relevance at 2 if much of the background is tangential. Cap Clarity at 3 if step boundaries are unclear or terminology is inconsistent.

A.1.4 Variant Evaluation Results

Audience	Length	Knowledge Comp.	Knowledge Rel.	Reasoning Acc.	Reasoning Rel.	Clarity
Graduate	Concise	4.95	4.84	4.99	5.00	4.87
Graduate	Long	5.00	4.53	5.00	4.99	4.48
High school	Concise	4.64	4.83	4.97	4.99	4.89
High school	Long	4.99	4.76	4.99	5.00	4.88

Table 4: Instruction quality scores (five-point Likert scale) across audience level and length variants.

B Results

B.1 Inference Settings

All models were evaluated using a consistent inference configuration. Generation employed a temperature of 0.7 and a top-p value of 0.95 across all tasks. For fairness, each model operated at its default context length (typically 8,192 tokens for smaller models). When the concatenated prompt and retrieved instructions exceeded the model's context window, the number of retrieved instructions was reduced from the top-5 matches (e.g., using the top-4) until the full prompt fit within the allowable limit. This ensures uniform decoding behavior while maintaining maximal use of retrieved guidance within each model's constraints.

B.2 Aggregate Performance Results

Task	Model	Instruction Acc.	Zero-Shot Acc.	Δ
MathQA				
LLaMA-3 1B Instruct	0.07	0.10	-0.03	
DeepSeek-R1 Qwen 1.5B	0.78	0.68	+0.10	
Qwen-3 1.7B Base	0.90	0.88	+0.01	
Gemma-2 2B IT	0.36	0.39	-0.02	
LLaMA-3 3B Instruct	0.15	0.11	+0.04	
Qwen-3 4B Base	0.91	0.91	+0.01	
Qwen-3 4B Instruct	0.91	0.90	+0.01	
DeepSeek-R1 Qwen 7B	0.88	0.88	+0.00	
Mistral 7B Instruct	0.35	0.22	+0.14	
DeepSeek-R1 LLaMA 8B	0.87	0.84	+0.03	
Gemma-2 9B IT	0.75	0.64	+0.10	
DeepSeek-R1 Qwen 14B	0.90	0.89	+0.02	
Qwen-3 14B Base	0.91	0.79	+0.12	
MedQA				
LLaMA-3 1B Instruct	0.19	0.30	-0.11	
DeepSeek-R1 Qwen 1.5B	0.31	0.21	+0.10	
Qwen-3 1.7B Base	0.60	0.45	+0.15	
Gemma-2 2B IT	0.32	0.36	-0.04	
LLaMA-3 3B Instruct	0.50	0.49	+0.01	
Qwen-3 4B Base	0.73	0.64	+0.08	
Qwen-3 4B Instruct	0.74	0.64	+0.10	
DeepSeek-R1 Qwen 7B	0.47	0.34	+0.12	
Mistral 7B Instruct	0.44	0.33	+0.11	
DeepSeek-R1 LLaMA 8B	0.60	0.51	+0.09	
Gemma-2 9B IT	0.54	0.37	+0.18	
DeepSeek-R1 Qwen 14B	0.74	0.68	+0.06	
Qwen-3 14B Base	0.83	0.73	+0.09	
MMLU (Law)				
LLaMA-3 1B Instruct	0.20	0.31	-0.11	
DeepSeek-R1 Qwen 1.5B	0.29	0.31	-0.01	
Qwen-3 1.7B Base	0.46	0.45	+0.01	
Gemma-2 2B IT	0.28	0.30	-0.02	
LLaMA-3 3B Instruct	0.38	0.34	+0.05	
Qwen-3 4B Base	0.57	0.47	+0.11	
Qwen-3 4B Instruct	0.54	0.49	+0.05	
DeepSeek-R1 Qwen 7B	0.36	0.37	-0.01	
Mistral 7B Instruct	0.38	0.22	+0.16	
DeepSeek-R1 LLaMA 8B	0.53	0.49	+0.04	
Gemma-2 9B IT	0.41	0.26	+0.14	
DeepSeek-R1 Qwen 14B	0.61	0.55	+0.07	
Qwen-3 14B Base	0.65	0.53	+0.12	

Table 5: **Aggregate performance for the High School Concise instruction variant.** Instruction retrieval consistently improves performance across tasks and models once capacity exceeds 3B parameters, with the largest gains on MedQA and MMLU Law. Δ denotes the difference between instruction and zero-shot accuracy.

C Ablation

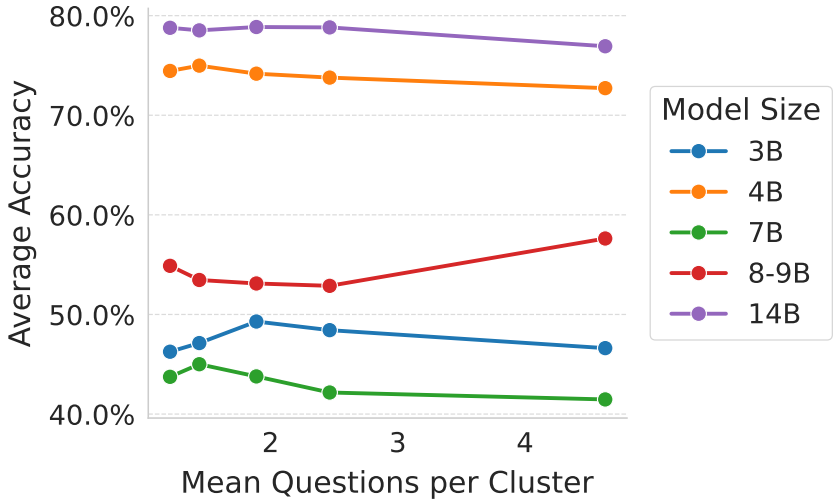

Figure 8: MedQA ablation results by model size. Instruction retrieval is relatively robust to clustering granularity for models 3B parameters. Brokenn down by model size

Model	Threshold 1	Threshold 2	Threshold 3	Threshold 4	Threshold 5
Avg. Cluster Size					
	1.21	1.44	1.89	2.46	4.63
Avg. Accuracy (%)					
deepseek_r1_llama_8b	62.0	61.1	59.1	59.6	57.6
deepseek_r1_qwen_14b	76.0	74.3	75.0	75.2	72.1
deepseek_r1_qwen_7b	45.3	45.0	40.8	38.7	35.8
gemma_2_9b_it	47.8	45.8	47.1	46.1	46.8
llama3_3b_instruct	46.1	47.1	49.3	48.4	46.6
mistral_7b_instruct	42.2	45.0	46.8	45.6	47.1
qwen3_14b_base	81.6	82.8	82.7	82.4	81.8
qwen3_4b_base	73.8	74.5	74.5	72.3	72.2
qwen3_4b_instruct	75.1	75.5	73.9	75.3	73.3

Table 6: Performance on MedQA across clustering thresholds by model. Each column corresponds to a clustering threshold, showing the average cluster size and model accuracy (%).

Model	Zero-Shot	RAG	Instructions	RAG Δ	Instr Δ
LLaMA-3 1B Instruct	0.30	0.22	0.19	-0.08	-0.11
DeepSeek-R1 Qwen 1.5B	0.21	0.26	0.31	+0.05	+0.10
Qwen-3 1.7B Base	0.45	0.53	0.60	+0.07	+0.15
Gemma-2 2B IT	0.36	0.27	0.32	-0.09	-0.04
LLaMA-3 3B Instruct	0.49	0.48	0.50	-0.01	+0.01
Qwen-3 4B Base	0.64	0.68	0.73	+0.04	+0.08
Qwen-3 4B Instruct	0.64	0.69	0.74	+0.05	+0.10
DeepSeek-R1 Qwen 7B	0.34	0.37	0.47	+0.03	+0.12
Mistral 7B Instruct	0.33	0.38	0.44	+0.05	+0.11
DeepSeek-R1 LLaMA 8B	0.51	0.56	0.60	+0.05	+0.09
Gemma-2 9B IT	0.37	0.40	0.54	+0.03	+0.18
DeepSeek-R1 Qwen 14B	0.68	0.69	0.74	+0.01	+0.06
Qwen-3 14B Base	0.73	0.78	0.83	+0.04	+0.09

Table 7: **MedQA: Zero-Shot vs RAG (Top-5 Passages) vs Instructions.** RAG retrieval from medical textbooks provides modest gains (+2 pp avg), while instruction retrieval achieves larger improvements (+7 pp avg). Both methods hurt performance on small models (1-2B parameters). Δ denotes the difference from zero-shot accuracy.

D Instruction Prompt Templates

````
# Question
{QUESTION}

# Required Output Format
```json
{{
  "reasoning": "<step-by-step analysis of the medical scenario and answer
      choices>",
  "final_answer": "A"
}}
```
````

Figure 9: Zero-shot Prompt

```
You are tasked with creating a **comprehensive, encouraging** instruction guide
    for **high school students** based on a set of examples of similar problems.
     Carefully examine these examples to identify common patterns, concepts, and
     problem-solving approaches. Your analysis should focus on extracting
    knowledge and reasoning patterns.

Write in **clear, encouraging language** that builds confidence for high school
    students. Provide **detailed explanations** to help students really
    understand. Your response must contain exactly these two sections with these
     exact headers:

## Background Knowledge
Present the essential concepts that directly help solve these problems. Include
    important patterns that help distinguish correct from incorrect answers. For
     each key point, indicate whether it strongly determines the answer when
    present, provides helpful support, or might mislead if given too much weight
    . **Use encouraging, detailed language with examples.** Connect new concepts
     to things students already know and explain why principles work the way
    they do.

## Reasoning Steps
Provide a detailed approach that works across examples, connecting each step to
    the background knowledge with encouraging explanations. Start by identifying
     what the question is asking and what key information or clues to look for.
    Explain how to apply the most important knowledge first, and when certain
    clues should override other considerations. Address how to weigh different
    types of evidence and handle situations where answers might seem similar.
    When relevant, explain how to tell apart commonly confused options by
    describing the key differences. End by stating what should determine the
    final choice and explain why. **Remember to explain the "why" behind each
    step to build understanding and confidence.** Here are the examples to
    analyze:

<examples>
{EXAMPLES}
</examples>

The instruction guide must work for every example provided.
```

Figure 12: High School Verbose Prompt Template

```
You are tasked with creating an instruction guide based on a set of examples of
    similar problems. Carefully examine these examples to identify common
    patterns, concepts, and problem-solving approaches. Your analysis should
    focus on extracting knowledge and reasoning patterns.

Consider the examples as a whole to understand their complexity and domain. Let
    the nature of the questions guide how simple or sophisticated and detailed
    your instructions should be. Write in clear, instructional language as if
    teaching someone how to solve this type of problem, using terminology and
    explanations that match the level demonstrated in the examples.

Your response must contain exactly these two sections with these exact headers:

## Background Knowledge
Present the essential principles, definitions, or rules that are directly
    relevant across the examples. Include important patterns that help
    distinguish correct from incorrect answers. For each key point, indicate
    whether it strongly determines the answer when present, provides helpful
    support, or might mislead if given too much weight. Use the language and
    concepts appropriate to the field and complexity level shown in the examples
    .

## Reasoning Steps
Provide a clear approach that works across examples, connecting each step to the
     background knowledge. Start by identifying what the question is asking and
    what key information or clues to look for. Explain how to apply the most
    important knowledge first, and when certain clues should override other
    considerations. Address how to weigh different types of evidence and handle
    situations where answers might seem similar. When relevant, explain how to
    tell apart commonly confused options. End by stating what should determine
    the final choice.

Here are the examples to analyze:

<examples>
{EXAMPLES}
</examples>

The instructions must work for every example provided.
```

Figure 10: Baseline Prompt Template

```
You are tasked with creating a **comprehensive, thorough** instruction guide for
     **graduate students** based on a set of examples of similar problems.
    Carefully examine these examples to identify common patterns, concepts, and
    problem-solving approaches. Your analysis should focus on extracting
    knowledge and reasoning patterns.

Write in precise, **graduate-level language** using **advanced terminology**
    appropriate for the domain. Provide **comprehensive coverage and detailed
    explanations**. Your response must contain exactly these two sections with
    these exact headers:

## Background Knowledge
Present comprehensive coverage of the essential principles, definitions, and
    domain knowledge directly relevant across most examples. Include important
    patterns that help distinguish correct from incorrect answers, even if they
    appear less frequently. For each key point, indicate whether it strongly
    determines the answer when present, provides helpful support, or might
    mislead if given too much weight. **Use precise graduate-level terminology
    .** Include detailed definitions with theoretical context, fundamental
    principles, complex relationships between concepts, and connections to
    broader theory when directly applicable.

## Reasoning Steps
Provide a comprehensive approach that works across examples, connecting each
    step to the background knowledge with detailed explanations. Start by
    identifying what the question is asking and what key information or clues to
     look for. Explain how to apply the most important knowledge first, and when
     certain clues should override other considerations. Address how to weigh
    different types of evidence and handle situations where answers might seem
    similar, explaining the theoretical foundations behind this hierarchy. When
    relevant, explain how to tell apart commonly confused options by
    comprehensively comparing the alternatives. Include multiple solution
    approaches when applicable, discuss trade-offs between methods, and address
    edge cases. End by stating what should determine the final choice with
```

```
You are tasked with creating a **quick, practical** instruction guide for **high
     school students** based on a set of examples of similar problems. Carefully
     examine these examples to identify common patterns, concepts, and problem-
    solving approaches. Your analysis should focus on extracting knowledge and
    reasoning patterns.

Write in **clear, simple language** that high school students can understand
    quickly. Keep your response **short and practical** - focus only on what
    students need to know to solve the problem. Your response must contain
    exactly these two sections with these exact headers:

## Background Knowledge
Present the essential concepts needed to solve these problems. Include important
     patterns that help distinguish correct from incorrect answers. For each key
     point, indicate whether it strongly determines the answer when present,
    provides helpful support, or might mislead if given too much weight. **Use
    simple language and keep explanations short and practical.**

## Reasoning Steps
Provide a clear approach that works across examples, connecting each step to the
     background knowledge. Start by identifying what the question is asking and
    what key information or clues to look for. Explain how to apply the most
    important knowledge first, and when certain clues should override other
    considerations. Address how to weigh different types of evidence and handle
    situations where answers might seem similar. When relevant, explain how to
    tell apart commonly confused options. End by stating what should determine
    the final choice. **Keep steps clear and practical.** Here are the examples
    to analyze:

<examples>
{EXAMPLES}
</examples>

The instruction guide must work for every example provided.
```

Figure 11: High School Concise Prompt Template

```
You are tasked with creating a **concise** instruction guide for **graduate-
    level students** based on a set of examples of similar problems. Carefully
    examine these examples to identify common patterns, concepts, and problem-
    solving approaches. Your analysis should focus on extracting knowledge and
    reasoning patterns.

Write in precise, **graduate-level language** using terminology appropriate for
    the domain. Keep your response **concise and focused on essential
    information only**. Your response must contain exactly these two sections
    with these exact headers:

## Background Knowledge
Present the essential principles, definitions, and domain knowledge directly
    relevant across most examples. Include important patterns that help
    distinguish correct from incorrect answers, even if they appear less
    frequently. For each key point, indicate whether it strongly determines the
    answer when present, provides helpful support, or might mislead if given too
     much weight. **Use precise graduate-level terminology.**

## Reasoning Steps
Provide a systematic approach that works across examples, connecting each step
    to the background knowledge. Start by identifying what the question is
    asking and what key information or clues to look for. Explain how to apply
    the most important knowledge first, and when certain clues should override
    other considerations. Address how to weigh different types of evidence and
    handle situations where answers might seem similar. When relevant, explain
    how to tell apart commonly confused options. End by stating what should
    determine the final choice. **Keep explanations concise but complete.** Here
     are the examples to analyze:

<examples>
{EXAMPLES}
</examples>

The instruction guide must work for every example provided.
```

Figure 13: Graduate Concise Prompt Template

```
 Question
{QUESTION}

# Retrieved Context
The following passages from medical textbooks may be relevant to answering this
    question.

{PASSAGES}

# Required Output Format
'''json
{{
  "reasoning": "<step-by-step analysis of the medical scenario and answer
      choices>",
  "final_answer": "A"
}}
'''
```

Figure 15: MedQA RAG baseline Prompt