

On-Policy Context Distillation

Stanford CS224N Custom Project

Simon Pritchard

Department of Statistics
Stanford University
skpritch@stanford.edu

Darynne Lee

Department of Computer Science
Stanford University
darynnel@stanford.edu

Shizhe He

Department of Computer Science
Stanford University
shizhehe@stanford.edu

Abstract

Large language models (LLMs) rely on in-context learning (ICL) for long-document question answering, an approach that is both compute and memory-intensive due to linear KV-cache scaling [1, 2]. The CARTRIDGES framework addresses this by distilling a document into a reusable prefix adapter via offline SELF-STUDY: a frozen base model generates synthetic QA pairs, and a KL-divergence objective trains the prefix to approximate full in-context behavior [3]. However, this offline approach is frozen in that the synthetic data is generated before learning begins, so the questions reflect the base model’s initial biases rather than the cartridge’s dynamically evolving understanding. We propose an on-policy SELF-STUDY strategy, which regenerates synthetic QA data per training iteration using the base model conditioned on the current cartridge prefix. The base LM weights remain frozen; only the prefix adapter is trained. Because the updated cartridge alters what the model attends to in the document, successive iterations produce questions that target the cartridge’s current failure modes rather than revisiting already-captured facts [4]. We evaluate on three documents from LONGBENCH-V2 [5] spanning 13k–310k tokens, comparing ICL, off-policy, and on-policy cartridges on held-out comprehension questions. On medium-length documents, on-policy cartridges surpass the off-policy baseline by 5.8 percentage points on held-out comprehension questions, with matched gains on long documents; the improvement coincides with a qualitative shift in generated questions from factual recall toward inference and cross-passage synthesis.

1 Key Information to include

- Mentor: Mirac Suzgun
- External Collaborators: None
- Late Days (if you are using any): 2

Team member	Late days remaining
Simon Pritchard	3
Darynne Lee	1
Shizhe He	3
Total pooled	7
Team size	3
Late days applied for the report	2

2 Introduction

Large language models (LLMs) are increasingly used to answer queries grounded in large text corpora such as codebases, legal filings, and scientific documents [6]. Historically, this entails loading the entire document into the model’s context window at inference time [7]. While effective, this approach entails high compute costs and performance degrades as the required context length increases [1, 2, 8]. The key-value (KV) cache used by Transformer attention scales linearly with the number of tokens, resulting in large memory footprints and reduced inference throughput for long-context queries [1, 2].

An alternative strategy is in distilling document knowledge into reusable model representations, allowing for document understanding without the need for ICL at inference time [9, 10, 3]. Under this paradigm, a model learns a compact representation through training that can be applied to diverse queries at a later date without maintaining the document in context. This aims to amortize long-context reasoning costs across future queries while retaining typical ICL reasoning abilities [3].

Recent work has explored learning representations from distilling on synthetically generated question/answer pairs [3, 11]. Under this paradigm, a base model first generates synthetic questions about the document, and a prefix adapter is trained to minimize the KL divergence between the model trained on this representation and the base model with ICL. This process, however, typically relies on static data synthesis: training questions are synthesized once by a frozen base model prior to training [3]. Since the base model never sees the cartridge, the resulting questions reflect the base model’s own understanding and what portions of the document it finds salient to continue querying, rather than those of the cartridge. The distribution fails to adapt to the cartridge’s dynamic understanding, theoretically leaving systematic gaps in the distilled representation [3, 4].

Here, we investigate whether long-document understanding is improved by on-policy data generation via context distillation [4, 3]. We modify the CARTRIDGES training loop: rather than generating all synthetic QA data from the frozen base model, we split training into K iterations and synthesize new QA data at each iteration, conditioning on the latest cartridge prefix [4, 11]. The base LM is frozen throughout; only the prefix adapter is trained. Therefore, “on-policy” refers to the evolving cartridge, not to updates of the base model weights. The total number of synthetic samples is constant, so performance differences are attributable causally to distributional shift. As the cartridge improves at representing a document region, the cartridge-conditioned model generates deeper questions on the given region, targeting inference and cross-passage reasoning rather than the cartridge repeatedly learning surface-level facts.

We evaluate this approach on long-context question answering, where a model must be able to parse up to hundreds of thousands of tokens [5]. After training, the document is represented through a learned prefix adapter; the model answers queries without loading the full document at inference [3, 9]. We compare three settings: in-context learning, offline SELF-STUDY distillation, and our proposed on-policy SELF-STUDY method. Experiments are conducted on documents from LongBench-V2 [12, 5].

We evaluate on three documents from LONGBENCH-V2 [5] spanning short (~ 13 k tokens), medium (~ 26 k tokens), and long (~ 310 k tokens) regimes. On the medium document, on-policy surpasses the off-policy baseline by 5.8pp (88.2% vs. 82.4%) after four iterations with an identical total data budget. The long document also surpasses off-policy (62.7% vs. 56.9%). Qualitatively, the synthetic questions shift from factual recall at early iterations to inference and cross-passage synthesis at later iterations, consistent with the hypothesis that on-policy generation tracks the cartridge’s evolving gaps. Our experiments address the following questions:

1. *Does on-policy SELF-STUDY recover more of the ICL upper bound than off-policy?* We train both off-policy and on-policy cartridges on the same documents with identical total data budgets and compare accuracy on held-out comprehension questions. On-policy surpasses off-policy on medium (+5.8pp) and long (+5.8pp) documents (Section 5.4).
2. *Is the gain from distributional shift or from data volume?* The total number of synthetic QA pairs is held constant across conditions (16k/32k/64k for short/medium/long), so the on-policy gains are attributable to the evolving question distribution rather than additional training signal (Section 5.4).

3. *How does the synthetic question distribution change across iterations?* At $K=0$, questions target surface-level facts; by $K=2$, they require multi-paragraph inference and characterization. Appendix C provides side-by-side examples.

3 Related Work

Context distillation and prefix tuning. Context distillation trains a model such that it "understands" enough of the context to respond to a diverse set of queries without employing ICL. This is typically affected by minimizing the KL divergence between prompted and unprompted outputs [13, 14]. Several methods for learning compressed context representations are as follows: gist tokens compress prompts into cacheable embeddings [15], AutoCompressor recursively summarizes segments into soft tokens [16], and prefix tuning [9] and adapters [10] condition frozen backbones on small trainable modules. The CARTRIDGES framework [3] employs this to full documents, training a prefix KV cache via a SELF-STUDY procedure in which the base model generates synthetic QA pairs and a KL-divergence objective distills the full-context model into the prefix adapter. Our work retains the prefix-adapter parameterization and KL-divergence objective of CARTRIDGES but replaces its static, one-shot data generation with an iterative on-policy loop.

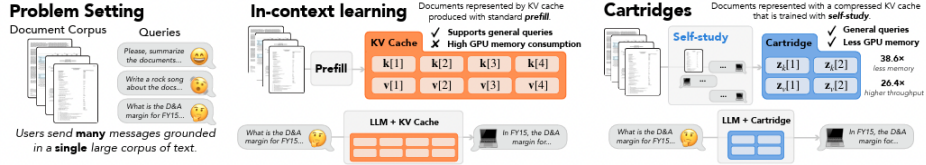


Figure 1: **The CARTRIDGES framework distills a long document into a reusable prefix adapter via SELF-STUDY [3].** A base model generates synthetic QA conversations about the document, and then a prefix KV cache is trained with a KL divergence objective such that the resulting conditioned model learns to approximate the behavior of the full ICL model. Upon inference, the learned prefix replaces the raw document, lowering the cost of ICL-based inference across many queries.

KV cache compression. Complementary to distillation, KV cache compression reduces the memory footprint of serving long contexts directly. Token-eviction methods such as H₂O [17] and SnapKV [18] prune low-attention tokens; quantization approaches like KVQuant [2] and KIVI [19] reduce bit-width; and learned retention gates [20] train per-token importance scores via distillation under strict memory budgets. Prompt-level compression methods such as LLMLingua [21] drop tokens before they enter the model. These approaches must be re-applied per document, whereas prefix adapters produce reusable representations that amortize cost across queries [3]. Our approach sidesteps per-query compression entirely by distilling the document into a fixed prefix that is computed once and reused, with on-policy training improving the quality of that prefix.

On-policy distillation and self-training. Standard knowledge distillation [22, 23] trains a student on fixed teacher outputs, creating a distribution mismatch at inference time. GKD [24] addresses this by training on student-generated sequences with teacher feedback, while MiniLLM [25] optimizes reverse KL on policy-generated outputs. In self-training, STAR [11] bootstraps reasoning by retraining on successful rationales, and privileged distillation [26] transfers knowledge unavailable at test time. We apply a similar on-policy principle: instead of regenerating *answers*, we regenerate the *questions* driving the distillation objective so that the data distribution tracks the cartridge’s evolving gaps. Unlike STAR, the base LM is frozen; the on-policy shift arises solely through the learned prefix. Where GKD and MiniLLM apply on-policy learning to the *output* side of distillation (student-generated answers), we apply it to the *input* side (student-generated questions), targeting the cartridge’s current failure modes while keeping the teacher’s full-context signal unchanged.

Test-time adaptation. Rather than learning reusable representations, test-time methods adapt per input. SIFT [27] selects fine-tuning data via transductive active learning to maximize information gain about the current prompt, and TTT layers [28] replace attention with per-sequence gradient updates. These share our motivation of aligning training signal with model state but operate at inference time

on individual inputs rather than amortizing across queries. Our method occupies a middle ground: like test-time approaches it conditions data selection on the model’s current state, but like offline distillation it produces a reusable artifact — a trained prefix — that serves many queries without repeated adaptation.

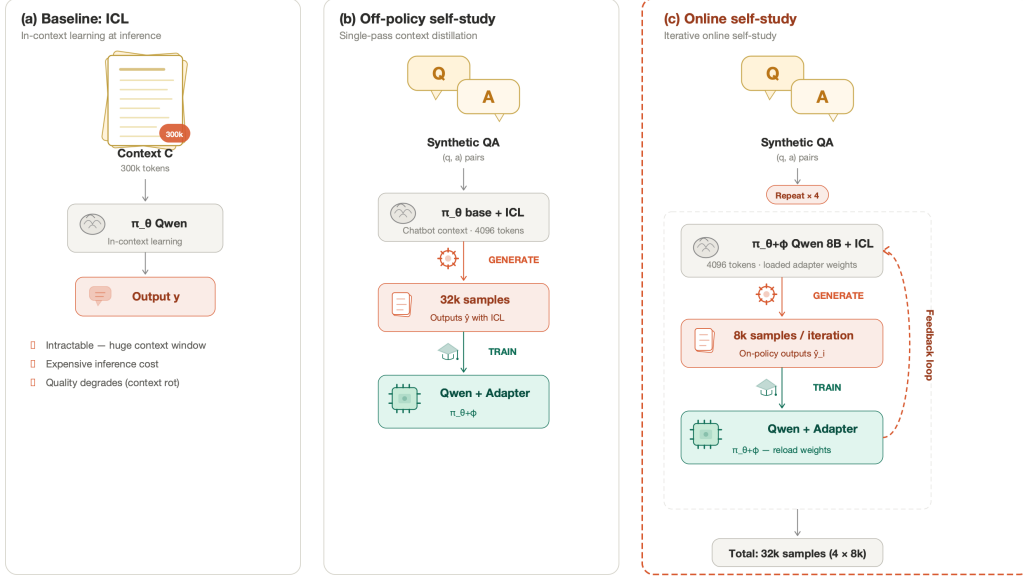


Figure 2: In **ICL-based inference (left)**, a base model ingests a document into its context at inference time to respond to queries. This has the downside of requiring massive context windows, necessitating high inference costs, and degrading performance as corpus size increases. In **off-policy SELF-STUDY (middle)**, a frozen base model synthesizes all QA pairs once before training begins; the data distribution remains fixed regardless of the cartridge evolving understanding. In **on-policy SELF-STUDY (right)**, the base model, conditioned on the current cartridge adapter, generates new QA pairs throughout training. The base LM weights are frozen in all settings; only the cartridge adapter is trained.

4 Approach

Baselines. We evaluate multiple baseline models on **LongBench-V2** in two modes: ICL (full document in the system prompt) and no-context (question only). The ICL baselines provide a notion on performance without distillation, while the no-context results are random guessing/reasoning based on the question alone.

On-policy context distillation. We modify the **CARTRIDGES** training loop at a single point: instead of generating all synthetic QA pairs once from the frozen base model, we split training into K iterations and regenerate the QA data at each iteration using the base model conditioned on the previous iteration’s cartridge prefix. The base LM weights remain frozen throughout; only the prefix adapter is trained. The distillation objective and cartridge parameterization are otherwise identical to **CARTRIDGES**.

At each iteration $k \in \{1, \dots, K\}$, three steps occur. First, the cartridge-conditioned model from the previous iteration (base model + cartridge_{k-1} , with cartridge_0 denoting the uninitialized cartridge) generates a fresh batch of synthetic QA pairs via the two-agent **SELF-STUDY** procedure described below. Second, the KL-divergence context-distillation objective is computed using the full-context model as teacher: the teacher is the base model with the full document in its context window, and the student is the base model conditioned on the current cartridge_k . The cartridge is updated to minimize this divergence. Third, the updated cartridge_k becomes the data-generating model for the next iteration.

This design separates two roles: the *teacher* that provides the distillation target always has access to the full document, ensuring the training signal is grounded in ground-truth context and does not compound errors across iterations. The *data-generating model* is the cartridge-conditioned model, whose questions adapt to what the cartridge has and has not yet learned. The total number of synthetic QA pairs is held constant across off-policy and on-policy conditions, so any performance difference is attributable to the distributional shift in the questions rather than data volume.

On-policy teacher model. As an ablation, we also evaluate a variant in which the teacher receives both the full document *and* the previous iteration’s cartridge (base model + full context + cartridge_{*k*−1}). The motivation is that on long documents where even full-context ICL struggles, the cartridge may help the teacher attend to relevant passages it would otherwise miss. The student remains the same (base model + cartridge_{*k*}). This variant adds KV-cache overhead at training time but may provide a richer distillation signal for documents that exceed the model’s effective attention span.

5 Experiments

5.1 Dataset

We evaluate on-policy SELF-STUDY LONGBENCH-V2: a long-document question/answer benchmark with 503 four-choice multiple-choice questions spanning a variety of domains [5]. We stratify questions into three token-length buckets, short ($\leq 20k$ tokens), medium (20–38k), and long (70–100k), estimated at 4 characters per token. It’s worth noting that this deviates from LONGBENCH-V2’s internal length stratification labels due to model context window constraints on our part. We sample 51 short, 84 medium, and 45 long documents (180 total) using a fixed random seed, maintaining a 1:2 easy-to-hard difficulty ratio within each bucket to control for potential sampling discrepancies across lengths.

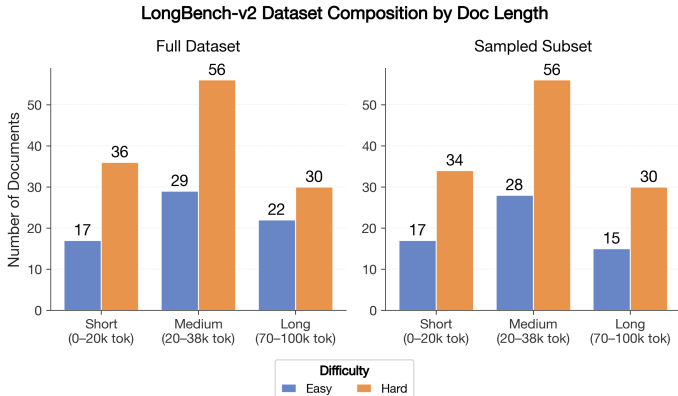


Figure 3: **LONGBENCH-V2 composition of short, medium, and long documents, stratified by ‘easy’ or ‘hard’ difficulty.** Overall LONGBENCH-V2 distribution (left) with easy document proportions (blue) and hard document proportions (orange). **Sampled LONGBENCH-V2 distribution (right).** Ratios of easy/hard docs were fixed across size categories to ensure sampling error didn’t introduce variance in baselines.

Context distillation. We select three documents from LONGBENCH-V2, spanning distinct token-length regimes: the novel *Anne of Green Gables* (short, $\sim 13k$ tokens), a collection of academic papers on reference-augmented 3D generation (medium, $\sim 26k$ tokens), and the novel *Moby-Dick* (long, $\sim 310k$ tokens). We want to select documents where our on-policy distillation has room to demonstrate advantages over the base model, so we employ our baseline evaluation results to identify certain documents on which the base model answers the original LONGBENCH-V2 question incorrectly under ICL and no-context inference paradigms.

SELF-STUDY synthesis. For each document, synthetic question/answer pairs are generated via a two-agent SELF-STUDY procedure [3]. A Model A and Model B are instantiated from an identical base model. In the on-policy setting, both models are conditioned on the current training iteration’s

cartridge (the base LM weights are, per usual, frozen — only the prefix evolves over training). At each iteration, a random contiguous chunk of the document (4,096–8,192 tokens) is selected and placed in both agents’ system prompts. Model A receives the document chunk along with one of four instructions, selected uniformly at random, each targeting a differing comprehension skill: detail retrieval (locating a specific fact buried in the passage), cross-passage synthesis (combining information from throughout the chunk), inference (deriving answers through some amount of reasoning), and discrimination (distinguishing easily confused entities or attending to qualifications). Model A synthesizes a single question; Model B then answers it with step-by-step reasoning founded on the same document chunk. Both agents generate with extended thinking enabled. Full seed prompts and system prompts are provided in Appendices A and B.

We scale the number of synthetic question/answer pairs with document length: 16k for the short document, 32k for medium, and 64k for the long strata. In the *off-policy* baseline, all samples are generated once by the frozen base model prior to training. In the *on-policy* variant, generation is split across $K=4$ equivalent iterations, with the SELF-STUDY procedure repeated each iteration using the base model conditioned on the latest cartridge prefix. The total number of synthetic QA pairs is held constant across condition sets to isolate the effects of on-policy regeneration versus simple data volume scaling.

5.2 Models

Baselines. We evaluate five models spanning a range of parameter counts and context-window sizes: LLAMA-3.1-8B (131k), QWEN3-30B-A3B (41k), QWEN3-8B (41k), QWEN3-32B (41k), KIMI-K2 (131k, MoE, ~32B activated), and KIMI-K2.5 (262k, MoE, ~32B activated). For the scaling analysis, both KIMI models are plotted at 32B (activated parameters per token) rather than total parameter count.

Context distillation. We train QWEN3-8B by freezing all base LM weights and training only a prefix KV-cache adapter of length 2,048 tokens. Training uses AdamW with learning rate 5×10^{-2} , $\epsilon = 10^{-8}$ and a 20-step linear warmup. The loss is KL divergence between the cartridge-conditioned student and the full-context teacher. Self-study conversations are packed into sequences of 8,192 tokens with top- $k=20$ logit truncation. We train for one epoch with a global batch size of 32 across $8 \times$ H100 GPUs using FSDP. For on-policy training, each step $k > 0$ initializes from the previous step’s checkpoint.

5.3 Evaluation

Baselines. Each model is evaluated in two modes per question: ICL, where the full document is provided in the system prompt, and no-context, where only the question and answer choices are presented. ICL evaluation is context-window gated; a model receives the document only when the estimated token count plus a 2k-token overhead fits within its context limit. In practice, both QWEN models (41k context) are excluded from ICL on long documents (minimum 70k tokens). No-context evaluation runs on all 180 questions for every model. We report mean accuracy and binomial standard error. All API calls use temperature 0.3 and answers are parsed from model outputs using pattern matching.

Cartridge evaluation. We evaluate each trained cartridge on a held-out set of 50 synthetic multiple-choice questions per document (150 total). For each document, we sample 10 random chunks, 4,096 tokens for short, 8,192 for medium, and 32,768 for long, and prompt CLAUDE OPUS 4.6 to generate 5 four-choice comprehension questions per chunk. Questions are constrained to require passage-level evidence and span factual recall, inference, and synthesis. These evaluation questions are generated once and held fixed across all experimental conditions.

5.4 Results

We first establish baselines quantifying the ICL–no-context gap that cartridges must close, then report on-policy distillation results and compute costs.

ICL headroom varies by document length. ICL outperforms no-context across all models, with the largest gains on medium documents (30+pp for KIMI models) and a moderate ~ 15 pp lift on long documents (Figure 4). This gap defines the *headroom* a cartridge can recover. Short documents show the smallest gap, suggesting base models already capture much of the content parametrically. Long documents, despite being far larger, yield only moderate ICL gains, indicating that attention struggles to exploit very long contexts even when they are available. Medium documents occupy a sweet spot: long enough to exceed parametric knowledge, short enough for attention to exploit effectively.

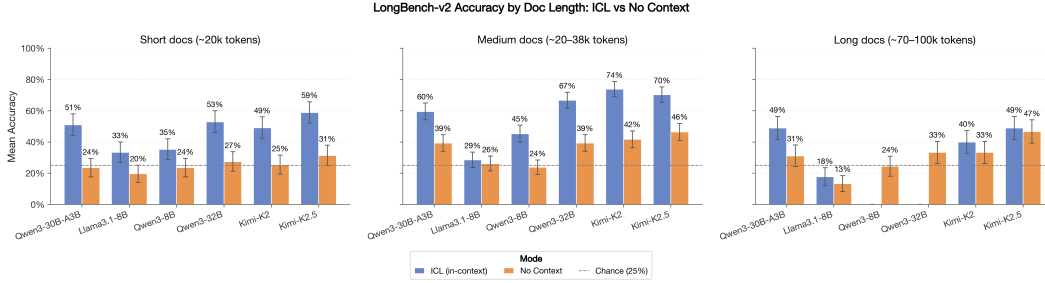


Figure 4: **ICL headroom is largest on medium-length documents.** Mean accuracy by model and mode (ICL vs. no-context) across token-length buckets: **(Left)** short (≤ 20 k), **(Middle)** medium (20–38k), **(Right)** long (70–100k). Error bars: binomial SE. Dashed line: 25% chance. QWEN3-8B/32B excluded from long ICL (context-window constraint).

Scaling model size yields diminishing returns. The pairwise scaling analysis (Figure 5) shows the steepest accuracy gains on medium documents, intermediate on long, and shallowest on short. This motivates distillation: rather than deploying a larger model, distill the document into a compact prefix that a smaller model can exploit.

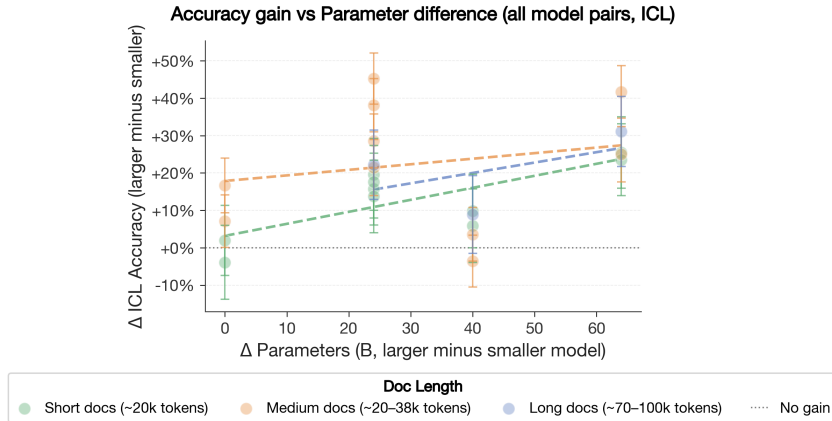


Figure 5: **Scaling gains across document length.** Each point is an ordered model pair (smaller $\rightarrow$ larger); MoE models plotted at activated parameters (~ 32 B). Long-document fit uses only 3 pairs (40k-context models excluded).

On-policy accuracy improves with iteration. Figure 6 shows cartridge accuracy across on-policy iterations for all three documents. The medium document shows the strongest on-policy gains: accuracy climbs from 64.7% at step 0 to 88.2% by step 3, surpassing the off-policy baseline (82.4%) by 5.8pp. The long document also surpasses off-policy, rising from 60.8% to 62.7% versus 56.9% off-policy. The short document is roughly flat at $\sim 72\%$ across iterations, below its off-policy baseline (80.4%). These results suggest that on-policy regeneration is most beneficial for medium-length documents, consistent with our baseline finding that this regime offers the largest ICL headroom.

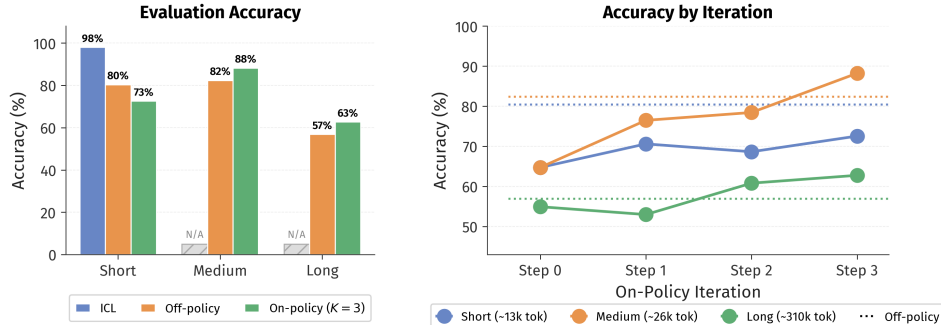


Figure 6: **On-policy cartridges improve their accuracy with iterations; medium and long surpass their off-policy baselines by step 3.** (Left) Accuracy by condition: ICL (full document in context), off-policy cartridge, and on-policy cartridge at $K=3$. ICL is unavailable for medium and long documents (doc length exceeds context window). On-policy context distillation marginally surpasses off-policy on medium (+5.8pp) and long (+5.8pp). (Right) On-policy accuracy per iteration (solid lines) versus off-policy baseline (dotted lines). Medium gains +23.5pp across iterations; long improves steadily over time; short is roughly flat at $\sim 72\%$ across training iterations.

On-policy generation shifts from recall to inference. The accuracy gains coincide with a qualitative shift in the synthetic QA distribution. At $K=0$, the frozen base model generates factual-recall questions targeting surface details (names, locations, direct quotes) that the cartridge captures early. By $K=2$, the cartridge-conditioned model generates inference and characterization questions requiring multi-paragraph synthesis, e.g., tracing a character’s evolving emotional arc through scattered dialogue cues. Appendix C provides side-by-side examples.

Directions not evaluated. Due to compute constraints, two planned experiments remain incomplete. First, we hypothesize that augmenting the teacher (model B) with the previous iteration’s cartridge (full context + cartridge_{k-1}) would most benefit the long document, where even the full-context teacher struggles to attend over 310k tokens and the cartridge could focus attention on underrepresented passages. Second, we expect on-policy cartridges to exhibit less catastrophic forgetting on MMLU than off-policy, since on-policy data stays closer to the model’s current distribution and avoids the covariate shift that drives forgetting in offline distillation [24].

Compute cost. On-policy training takes $\sim 4\times$ the wall-clock time of off-policy (Table 1). The per-iteration cost is comparable; the overhead is sequential checkpointing between iterations, which an asynchronous pipeline would largely eliminate.

Condition	Samples	GPUs	Wall-clock	GPU-hours
Off-policy ($K=1$)	16k	16	3m 06s	0.83
On-policy ($K=4$)	$4\times 4k$	16	$4\times 3m = 12m$	3.20

Table 1: **Training cost for the short document ($\sim 13k$ tokens) on 2×8 H100s.** Total sample budget is identical (16k); on-policy splits it across 4 iterations.

6 Conclusion

We proposed on-policy self-study for context distillation, regenerating synthetic QA data at each iteration using the current cartridge prefix rather than a frozen base model. On medium- and long-length documents, on-policy cartridges surpass the off-policy baseline by 5.8pp each; on short documents, where the base model already captures most relevant content, on-policy underperforms off-policy. The improvement coincides with a shift in generated questions from factual recall toward inference and cross-passage synthesis, suggesting the data distribution adapts to the cartridge’s evolving capabilities. The on-policy loop incurs roughly $4\times$ generation wall-clock time, though

per-iteration cost is comparable. Our results indicate that even a mild form of on-policy learning—where only a prefix adapter evolves—can meaningfully improve distilled document representations, particularly in the medium-length regime where ICL headroom is largest.

Team contributions (Required for multi-person team)

Darynne Lee was responsible for a lot of the project inspiration, framing, and background research. Simon Pritchard made the Longbench_v2 data pipeline and was responsible for benchmarking the baselines. Shizhe He was responsible for most of (in conjunction with Darynne Lee) the coding relating to the actual on-policy context distillation implementation. All wrote the report.

Acknowledgements

We thank our project mentor, Mirac Suzgun, for his guidance and feedback throughout this work. We are grateful to the CS 224N teaching team for their support and mentorship, and especially for their understanding and flexibility during a difficult time for our team. We also thank Engram Labs for generously providing compute resources used for model training and evaluation.

References

- [1] Kwon et al. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.
- [2] Hooper et al. Towards 10 million context length: KV cache compression for large language model inference with KVQuant. In *Advances in Neural Information Processing Systems*, 2024.
- [3] Eyuboglu et al. Cartridges: Lightweight and general-purpose long context representations via self-study, 2025.
- [4] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. <https://thinkingmachines.ai/blog/on-policy-distillation>.
- [5] Bai et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.
- [6] Lewis et al. Retrieval-augmented generation for knowledge-intensive NLP tasks, 2020.
- [7] Brown et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.
- [8] Kelly Hong, Anton Troynikov, and Jeff Huber. Context rot: How increasing input tokens impacts llm performance. Technical report, Chroma, July 2025.
- [9] Li et al. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.
- [10] Houshy et al. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799, 2019.
- [11] Zelikman et al. STaR: Bootstrapping reasoning with reasoning, 2022.
- [12] Yang et al. Qwen3 technical report, 2025.
- [13] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.
- [14] Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

- [15] Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *Advances in Neural Information Processing Systems*, 2023.
- [16] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.
- [17] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H₂O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, 2023.
- [18] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, 2024.
- [19] Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. *arXiv preprint arXiv:2402.02750*, 2024.
- [20] Adrian Łańcucki et al. TRIM-KV: Cache what lasts: Token retention for memory-bounded KV cache in LLMs. *arXiv preprint arXiv:2512.03324*, 2025.
- [21] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023.
- [22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- [23] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, 2016.
- [24] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*, 2024.
- [25] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.
- [26] Emiliano Penaloza et al. Privileged information distillation for language models, 2026.
- [27] Jonas Hübötter, Sascha Bongni, Ido Hakimi, and Andreas Krause. Efficiently learning at test-time: Active fine-tuning of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [28] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Kumbhar, et al. Learning to (learn at test time): RNNs with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

A SELF-STUDY Seed Prompts

At each SELF-STUDY step, Model A receives a random contiguous chunk of the document along with one of the following four seed instructions, selected uniformly at random. Each targets a distinct comprehension skill. The shared preamble (“You are provided with a passage from a long document. . .”) is included in every prompt and omitted here for brevity.

Detail Retrieval

Your goal is to generate a question whose answer is a specific fact, value, name, or detail that appears in the passage, but not in an obvious position like a heading, abstract, or opening paragraph. The answer should require locating the relevant sentence or data point among a large volume of surrounding material. The answer should be precise enough that there is only one correct response (a number, a name, a date, a short phrase). Do not ask questions whose answer could be guessed from general knowledge or from skimming the passage.

Be creative, don't just ask about the most prominent topic. Look for details that are mentioned once, buried in a list, embedded in a parenthetical, or stated in passing. Make sure the question is answerable from the passage. Keep the question reasonably brief.

Cross-Passage Synthesis

Your goal is to generate a question that can only be answered by combining information from at least two separate parts of the passage. Someone who has only read one paragraph, one section, or one segment should not be able to answer correctly.

Good patterns include: comparing a value or claim from one part with a value or claim from another, explaining how something introduced early interacts with an outcome described later, identifying a contradiction or evolution across the passage, or computing something that requires gathering numbers from different locations.

The question should have a definite, concrete answer, not "compare and contrast" or "discuss," but something with a specific resolution. Make sure the question is answerable from the passage. Be creative, don't just ask about the most obvious cross-references. Keep the question reasonably brief.

Inference

Your goal is to generate a question whose answer is not directly stated anywhere in the passage, but can be logically derived from what is stated. The reader must reason, not just search.

Good patterns include: inferring causality from correlated observations, identifying what must be true given multiple stated constraints, determining an unstated assumption behind an argument, computing a derived quantity from separately provided numbers, or drawing a conclusion that follows from the evidence but is never spelled out.

The question should have a single defensible answer, not an open-ended interpretation. Do not ask questions that require outside knowledge, the reasoning should be grounded entirely in the passage. Make sure the question is answerable from the passage. Be creative, look for non-obvious inferences. Keep the question reasonably brief.

Discrimination

Your goal is to generate a question that a careless or hasty reader would get wrong. Target something in the passage where a surface-level reading leads to one answer, but careful attention to detail leads to a different one.

Good targets include: easily confused entities (similar names, adjacent rows in a table, related but distinct concepts), values that are revised, corrected, or superseded later in the passage, claims that come with qualifications, exceptions, or caveats, and distinctions that require reading a full paragraph rather than just the first sentence.

The question itself should sound straightforward, the difficulty should come from the passage, not from a trick or convoluted phrasing. Make sure the question is answerable from the passage. Be creative, don't just target the most obvious potential confusion. Keep the question reasonably brief.

B System Prompts

In addition to the seed instructions (Appendix A), each agent receives a system prompt that frames its role and provides the document chunk. The same chunk is placed in both prompts so Model A and Model B reason over identical content.

Model A System Prompt (Questioner)

You are writing a comprehension question about the content below.

<content> {document chunk} </content>

Write one question that requires carefully reading the content above to answer. The question should be specific, have a definite answer, and sound like something a person would naturally ask, not like a test question. Do not provide the answer.

Model B System Prompt (Answerer)

You have access to the following content:

<content> {document chunk} </content>

The user will ask a question about this content. Answer it thoroughly using evidence from the content. Quote or reference specific parts to support your answer. If the answer requires combining information from multiple places, show how the pieces fit together. If you cannot answer from the content alone, say so.

C On-Policy QA Examples

The following examples are drawn from the short document (*Anne of Green Gables*, ~13k tokens). Both use the same document chunk covering Chapters I–III. The $K=0$ example is generated by the frozen base model (off-policy); the $K=2$ example is generated by the base model conditioned on the cartridge trained through two iterations (on-policy).

Iteration $K=0$ (off-policy)

Generated question: *What is the name of the orphan asylum mentioned in the passage?*

Generated answer (excerpt): The passage does not explicitly mention the name of the orphan asylum. However, it references Mrs. Alexander Spencer as the individual who was responsible for bringing the child (Anne Shirley) from the asylum. The text states: “Mrs. Spencer came off the train with that girl and gave her into my charge. Said you and your sister were adopting her from an orphan asylum. . .” While the asylum itself is not named, the narrative implies that it is a fictional or unspecified institution. . . . Thus, the name of the orphan asylum is not provided in the text.

Iteration $K=2$ (on-policy)

Generated question: *How did Matthew feel about the child’s talkative nature?*

Generated answer (excerpt): Matthew initially felt a mix of surprise and discomfort, but he gradually came to appreciate her lively conversation. He was “much to his own surprise” enjoying himself. He noted that he “liked talkative people when they were willing to do the talking themselves and did not expect him to keep up his end of it.” Despite finding her “brisk mental processes” challenging, he acknowledged that he “kind of liked her chatter” and encouraged her to continue, saying “Oh, you can talk as much as you like. I don’t mind.”

At $K=0$, the model asks a factual-recall question whose answer is not even present in the text, a dead-end that provides no useful training signal. At $K=2$, the cartridge-conditioned model generates an inference question requiring the reader to synthesize Matthew’s reactions across several paragraphs of dialogue. The answer demonstrates grounded, multi-evidence reasoning, the kind of comprehension behavior the distillation objective is designed to transfer into the cartridge.