Parallel Context-of-Experts Decoding for Retrieval Augmented Generation

Giulio Corallo
SAP Labs, France
EURECOM, France
giulio.corallo@sap.com

Paolo Papotti
EURECOM, France
papotti@eurecom.fr

Abstract

Retrieval Augmented Generation faces a trade-off: concatenating documents in a long prompt enables multi-document reasoning but creates prefill bottlenecks, while encoding document KV caches separately offers speed but breaks cross-document interaction. We propose *Parallel Context-of-Experts Decoding* (PCED), a training-free framework that shifts evidence aggregation from the attention mechanism to the decoding. PCED treats retrieved documents as isolated "experts", synchronizing their predictions via a retrieval-aware extension of context-aware decoding. This approach recovers cross-document reasoning capabilities without constructing a shared attention across documents.[1]

1 Introduction

Retrieval Augmented Generation (RAG) augments language models with external corpora to improve factuality and reduce hallucinations (Lewis et al., 2020; Gao et al., 2023; Fan et al., 2024). However, standard pipelines concatenate many retrieved documents into a single *long context* prompt, making inference dominated by prefill latency (Kwon et al., 2023; Zhong et al., 2024; Cheng et al., 2025). Additionally, long contexts increase reasoning failures, as models often struggle to integrate evidence spread across multiple documents (Liu et al., 2024).

Parallel KV cache encoding mitigates prefill cost by encoding retrieved documents independently and reusing their cached states at inference time (Yang et al., 2025b,c). However, removing cross-document attention during encoding can substantially degrade performance on multi-hop and reasoning-intensive queries (Yao et al., 2025).

We propose **Parallel Context-of-Experts Decoding** (PCED), a training-free framework that shifts document aggregation from attention to decoding. As depicted in Figure 1, at each generation step, PCED treats each document as a separate "expert", which proposes a next-token distribution from its own KV cache, and then weights the best-supported token so evidence can be *efficiently* aggregated across documents without building a joint attention context. Concretely, our contribution is the combination of (1) *parallel* per-document experts with modular KV caches (2) *token-level expert switching* to recover cross-document reasoning via dynamic expert selection *at every token step* without shared attention; and (3) *retrieval-integrated priors* that inject scalar scores into the contrastive decoding to gate noise from irrelevant experts. On benchmarks like LOFT and LongBench, PCED outperforms prior parallel methods by up to 70 points, often matching or exceeding *long context* baselines. Furthermore, it delivers over a $180\times$ speedup in time-to-first-token, maintaining substantial efficiency gains even when caches are built on the fly via parallel document encoding.

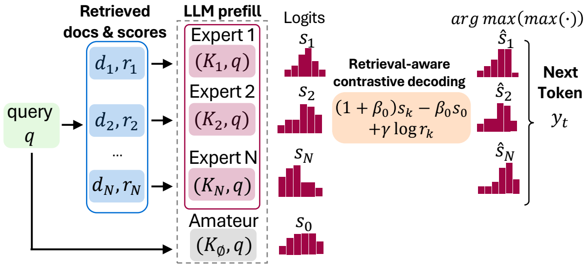

Figure 1: Parallel Context-of-Experts Decoding (PCED) runs one expert per retrieved document (and a no-context, amateur prior) in parallel and chooses each next token based on retrieval support, enabling evidence to be stitched across documents without joint attention.

2 Related Work

We position our work at the intersection of (1) KV caching for parallel prefill, (2) cross-document interaction recovery under independent KV caches,

[1] Code is available at https://github.com/giulio98/pced.

Findings of the Association for Computational Linguistics: ACL 2026, pages 32666–32676
July 2-7, 2026 ©2026 Association for Computational Linguistics

and (3) context-aware decoding.

Parallel encoding eliminates prefill cost by precomputing *offline* per-document KV caches that can be retrieved at inference time. Prior work includes training-free masking for blockwise/parallel attention (Ratner et al., 2023), fine-tuning to mitigate quality degradation under blocked attention (Ma et al., 2025), and interfaces that decouple document encoding from generation (Yen et al., 2024). Systems work integrates KV cache retrieval into RAG pipelines (Lu et al., 2025). These approaches assume documents as independently encodable, while we study how to aggregate evidence across multiple cached documents at inference.

Cache merging techniques encode documents independently and then aim to restore the cross-document attention, as simply concatenating per-document KV caches does not recover it (Yao et al., 2025). Recent methods achieve this via selective recomputation at merging (Yao et al., 2025), learned bridging tokens for inter-document interactions (Yang et al., 2025b), or training-free alignment to approximate sequential attention (APE) (Yang et al., 2025c). Our work preserves per-document modularity while enabling effective cross-document reasoning.

Context-aware decoding (CAD) (Shi et al., 2024) improves faithfulness by shifting probability mass toward tokens supported by context; it is related to contrastive decoding (Li et al., 2023) and classifier-free guidance in diffusion models (Ho and Salimans, 2021). However, most CAD formulations assume a single supportive context that defines the conditional distribution. DvD (Jin et al., 2024) extends CAD to multiple documents but collapses them into a single input sequence, which conflicts with per-document KV cache reuse, where documents must be encoded separately. Recent work explores aggregating retrieved documents at the *logit level*, e.g., entropy-based decoding for RAG (Qiu et al., 2025). These methods construct a *soft ensemble* over document-conditioned distributions, weighting experts by internal confidence. In contrast, PCED replaces soft mixtures with discrete, token-level max-expert routing. As shown in Table 9, this hard-switching strategy is particularly effective for multi-hop reasoning. Moreover, rather than relying on internal uncertainty, PCED uses external retrieval and reranker scores to suppress distractors, remaining robust even with large candidate sets where soft ensembles degrade (Table 4).

3 Methodology

We introduce Parallel Context-of-Experts Decoding (PCED), a training-free framework for scalable and faithful multi-document generation. RAG pipelines typically employ a two-stage process: retrieving candidate documents using embeddings to maximize *recall*, followed by a cross-encoder reranker to reorder candidates and maximize *precision*. Crucially, the scalar relevance scores produced during these stages are used only for document selection and then discarded. We argue that this discards valuable evidence about how strongly each document should be trusted during decoding. PCED converts these scores into *a document-level prior that controls how much each expert influences the next-token distribution*, via a retrieval-aware contrastive decoding criterion adapted to independently encoded document experts.

Offline KV cache preparation. Following prior cache-augmented generation work (Chan et al., 2025; Lu et al., 2025; Yang et al., 2025c; Jin et al., 2025), we assume a datastore $\mathcal{DB}$ over a corpus $\mathcal{D}$ that stores, for each document d_i, an embedding $\mathbf{e}_i$ for retrieval and its precomputed KV cache $\mathbf{K}_i$:

$$\mathcal{DB} = \{(d_i, \mathbf{e}_i, \mathbf{K}_i)\}_{i=1}^{|\mathcal{D}|}. \quad (1)$$

Precomputed KV caches are optional and serve purely as a systems optimization. PCED applies unchanged when document caches are constructed on the fly at inference time; cache reuse yields substantial efficiency gains, and parallel expert encoding provides speedups in both regimes.

Retrieval and relevance scoring. Given a query q, we retrieve the top-N documents and obtain retrieval scores $\mathbf{r}^{\text{ret}} = (r_1^{\text{ret}}, \ldots, r_N^{\text{ret}})$. We then rerank these documents with a cross-encoder, producing reranker scores $\mathbf{r}^{\text{rer}} = (r_1^{\text{rer}}, \ldots, r_N^{\text{rer}})$. We map both score sets to the range $[0, 1)$. Since r^{ret} primarily reflects recall and r^{rer} precision, we fuse them into a single per-document relevance score via the harmonic mean $r_k = \frac{2\, r_k^{\text{ret}}\, r_k^{\text{rer}}}{r_k^{\text{ret}} + r_k^{\text{rer}}}, k \in \{1, \ldots, N\}$.

Parallel Context-of-Experts. As depicted in Figure 1, PCED operates on $N{+}1$ parallel streams (*experts*) in a single batched forward pass: one **amateur** expert with an empty cache $\mathbf{K}_0 = \emptyset$ (model prior) and N **contextual** experts, one per retrieved document, with caches $\mathbf{K}_{1:N}$ and associated relevance scores $r_{1:N}$. Given batch $\mathcal{B} = \{\mathbf{K}_k\}_{k=0}^{N}$, processing the query q updates all experts' caches in parallel. At each step, this yields per-expert logits $s_k \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$.

Retrieval-aware contrastive decoding. For each contextual expert $k \in \{1, \ldots, N\}$, we use context-aware decoding to calibrate its predictions against the amateur s_0 and adapt it to a multi-expert setting by incorporating a retrieval-based prior:

$$\hat{s}_k = \underbrace{(1+\beta_0)\, s_k - \beta_0\, s_0}_{\substack{\text{Contrastive}\\ \text{decoding}}} + \underbrace{\gamma \log r_k}_{\substack{\text{Retrieval}\\ \text{prior}}} \tag{2}$$

Here, β_0 controls contrast strength between amateur and expert, and γ controls retrieval gating. We compute β_0 dynamically as in AdaCAD (Wang et al., 2025) for the *first* generated token and keep it fixed thereafter. We empirically set $\gamma = 2.5$ for all experiments (ablations in Appendix C.1 for β, C.2 for γ). Finally, the next token y_t is the one with the highest score among all experts' candidates.

$$y_t = \arg\max_{v \in \mathcal{V}} \left(\max_{k \in \{1,\ldots,N\}} \hat{s}_k(v) \right) \tag{3}$$

The chosen token is appended to the **shared generation history** for all experts at each step.

4 Experimental Setup

We test PCED on RAG, In-Context Learning (ICL), and long-context QA with distractors. For all methods, we fix the LLM, prompts, and retrieved candidates; varying only how context is incorporated.

Datasets and Metrics. We use the LOFT benchmark (Lee et al., 2024) for RAG and ICL. For each query, we use the benchmark-provided passages, rerank them with `bge-reranker-v2-m3`, and keep the top-90 documents; all methods and baselines then consume this same retrieved evidence. Performance is measured via *Subspan Exact Match* for RAG tasks and *Exact Match* for ICL tasks. We also evaluate on the query-focused LongBench subsets (Bai et al., 2024) using official metrics. To test robustness to irrelevant context, we concatenate the *gold* document with K=2 uniformly sampled *distractors* from other test samples, keeping the corpus-in-context baseline under 128k tokens.

LLMs. We report main results with MISTRAL-NEMO-13B-INSTRUCT (Mistral AI, 2024) and LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024), and LongBench results with QWEN3-8B (Yang et al., 2025a) extended to 128k tokens with YARN (Peng et al., 2024). Decoding is greedy for all methods to enable deterministic and fair paired comparisons. PCED only modifies the per-step logits, however, so standard sampling strategies (e.g., nucleus sampling (Holtzman et al., 2020)) can be applied on top of the aggregated distribution.

PCED variants. We evaluate three scoring variants: *Sparse*, *Dense*, and *ColBERT*. The set of retrieved documents is identical for all methods. These variants differ only in the relevance signal r_k extracted from `bge-m3` to weight experts in Eq. 2.

Baselines. We compare against three baseline families. Standard concatenation (*Corpus in Ctx*) conditions on either the **Single** top-1 document retrieved or **All** retrieved documents in a single prompt (e.g., top-90 for LOFT). *KV cache merging* (**APE**), prefills each document independently and merges the resulting KV caches. *Agentic aggregation* (**MAPREDUCE**) performs per-document summarization (map) followed by a final QA aggregation step (reduce) (Zhou et al., 2025).

5 Results and Discussion

We analyze our results around three main themes: (1) multi-document RAG and ICL with many candidate documents/exemplars, (2) single-document with query-focused understanding and generation tasks (including QA, summarization, code completion, and few-shot inference), and (3) efficiency.

Cross-Document Reasoning Emerges at Decode Time. In Table 1, PCED consistently outperforms KV cache merging (APE) in QA benchmarks that require aggregating evidence across multiple documents (e.g., HOTPOTQA, MUSIQUE, QAMPARI, QUEST), and in ICL settings where exemplars must be used jointly. For instance, on LLAMA-3.1-8B QAMPARI, PCED improves from 7 (APE) to 77 (PCED-Sparse), and yields up to +23 points over MAPREDUCE (e.g., HOTPOTQA). Moreover, PCED variants often match or exceed full-context concatenation: PCED-Dense outperforms Corpus in Ctx (All) in **10/16** settings despite encoding each document independently. These results suggest that much of the benefit of cross-document interaction can be recovered at *decode time*.

Figure 2 illustrates this mechanism: to resolve a multi-hop query, the model first locks onto an expert containing the bridging entity, then pivots to a second expert for the final answer. This is consistent with the model relying on different experts

Table 1: **Main results on RAG and ICL benchmarks.** We compare PCED equipped with Sparse, Dense, or ColBERT experts, against KV merging (APE), agentic (MAPREDUCE), and standard concatenation baselines. *Corpus in Ctx (All)* is the baseline with all retrieved candidates in context.

(a) MISTRAL-NEMO-13B-INSTRUCT

		KV Merge	Agentic	PCED			Corpus in Ctx	
Task	Dataset	APE	MapRed.	Sparse	Dense	ColBERT	Single	All
RAG	HOTPOTQA	27.0	56.0	65.0	**66.0**	66.0	54.0	64.0
	MUSIQUE	11.0	26.0	**36.0**	34.0	35.0	17.0	28.0
	NQ	38.0	62.0	80.0	**81.0**	81.0	60.0	76.0
	QAMPARI	7.0	**85.0**	75.0	71.0	71.0	75.0	74.0
	QUEST	1.0	42.0	**55.0**	54.0	54.0	38.0	25.0
ICL	Web	58.9	42.2	61.1	**62.2**	62.2	35.6	61.1
	Tracking7	6.7	**13.3**	7.8	7.8	7.8	10.0	6.7
	Date	40.0	55.6	57.8	**57.8**	57.8	57.8	54.4

(b) LLAMA-3.1-8B-INSTRUCT

		KV Merge	Agentic	PCED			Corpus in Ctx	
Task	Dataset	APE	MapRed.	Sparse	Dense	ColBERT	Single	All
RAG	HOTPOTQA	16.0	41.0	64.0	**64.0**	64.0	49.0	73.0
	MUSIQUE	4.0	8.0	14.0	**21.0**	21.0	7.0	28.0
	NQ	9.0	51.0	83.0	**85.0**	85.0	58.0	82.0
	QAMPARI	7.0	68.0	**77.0**	76.0	76.0	72.0	89.0
	QUEST	0.0	41.0	**45.0**	40.0	40.0	39.0	54.0
ICL	Web	61.1	56.7	62.2	**64.4**	63.3	57.8	57.8
	Tracking7	3.3	**13.3**	11.1	11.1	11.1	11.1	7.8
	Date	0.0	44.4	**53.3**	47.8	48.9	51.1	53.3

Table 2: **Results on LongBench using QWEN3-8B.** PCED against the full-context baseline Corpus in Ctx (All).

	Single-Doc QA			Multi-Doc QA			Summ.	Few-Shot	Code
Method	NARQA	QASPER	MULTIF	HOTPOT	2WIKI	MUSIQUE	QMSUM	TRIVIAQA	REPOB-P
Corpus in Ctx (All)	21.1	25.2	52.8	56.3	44.2	25.3	22.0	84.0	51.1
PCED-Sparse	25.1	24.2	**53.0**	62.1	49.4	**33.4**	22.7	**88.8**	59.7
PCED-Dense	**25.4**	**25.7**	52.6	**62.6**	**49.4**	33.3	**22.9**	88.2	**60.1**
PCED-ColBERT	25.4	25.7	52.6	62.6	49.4	33.3	22.9	88.2	60.1

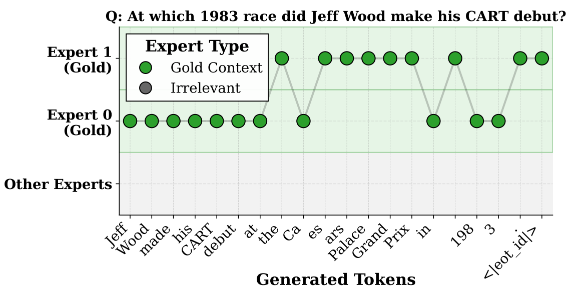

Figure 2: HOTPOTQA expert trace. Green dots illustrate the model hopping between multiple gold documents.

across generation steps. By appending the chosen token to *all* experts' shared generation histories, PCED effectively stitches evidence across isolated documents without shared attention. Comparing against MAPREDUCE, PCED-Dense wins in 12 out of 16 settings (Table 1). MAPREDUCE prevails where global compression suffices (QAMPARI, TRACKING7), but on evidence-sensitive tasks (HOTPOTQA, MUSIQUE, NQ) PCED benefits from direct access to raw documents rather than a lossy summarization bottleneck. The gap is also computational: MAPREDUCE requires $N+1$ sequential LLM calls (per-document summaries plus aggregation), whereas PCED aggregates in a single decoding pass over parallel experts, yielding a stronger overall accuracy–efficiency tradeoff. Paired approximate randomization tests on aligned example-level test-set scores (Table 3) corroborate these trends. PCED-Dense significantly outperforms MAPREDUCE on both HOTPOTQA (Δ=+0.22, p<10^{-3}) and NQ (Δ=+0.34, p<10^{-4}). Against the full-context baseline, the differences are not statistically significant (p=0.11 and p=0.45), suggesting that PCED competes with full-context reasoning despite encoding each document independently.

Table 3: **Paired statistical validation** using two-sided paired approximate randomization tests on aligned example-level test-set scores. Δ denotes the mean paired difference (PCED-Dense minus baseline). Confidence intervals are paired bootstrap 95% CIs.

Task	**Baseline**	Δ	**95% CI**	**p-value**
HOTPOTQA	MAPREDUCE	+0.22	[+0.11, +0.33]	3×10^{-4}
HOTPOTQA	Corpus in Ctx	−0.09	[−0.19, +0.01]	0.11
NQ	MAPREDUCE	+0.34	[+0.23, +0.45]	5×10^{-5}
NQ	Corpus in Ctx	+0.02	[−0.02, +0.08]	0.45

Table 4: **Robustness to large candidate pools on HOTPOTQA (LLAMA-3.1-8B).**

Method	$k=4$	$k=8$	$k=16$	$k=32$	$k=64$
CLeHe	46	33	24	18	14
LeEns	48	36	27	19	15
PCED-Dense	**64**	**64**	**64**	**64**	**64**

Table 4 further compares PCED with entropy-based logit-level aggregation methods (Qiu et al., 2025) under identical conditions (same model, prompts, and retrieved documents). As the candidate pool grows from k=4 to k=64, soft-ensemble

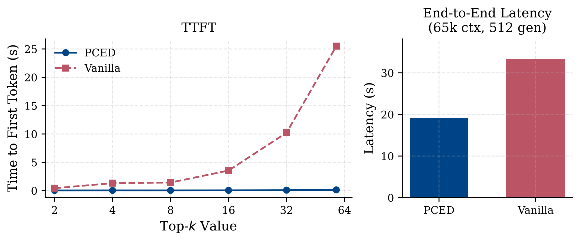

Figure 3: **Latency Benchmarks.** Comparison of TTFT scalability across Top-k values (left) and total end-to-end latency with 65k context (right).

approaches (CLeHe, LeEns) degrade sharply dropping from 46–48 to 14–15 on HOTPOTQA because entropy-weighted mixtures cannot suppress the accumulating distractor signal. PCED remains flat at 64 across all pool sizes, confirming that the combination of retrieval-prior gating and hard max-expert switching is critical for robustness in high-noise, multi-hop settings.

Less Noise, More Accuracy. PCED also improves performance on tasks where the answer is primarily supported by a *single* document, but must be recovered from a large candidate pool. In these settings, full-context concatenation can degrade because relevant evidence is diluted by many near-miss documents and distractors, making attention noisier. By contrast, PCED isolates evidence by treating each document as an independent expert and explicitly emphasizing per-document relevance via retrieval-aware contrastive decoding (Eq. 2), which downweights irrelevant experts. Table 1 shows that this yields strong gains on NQ under LOFT: with LLAMA, PCED-Dense improves from 58 (Corpus in Ctx Single) and 82 (All) to 85, similarly with MISTRAL from 60 (Single) and 76 (All) to 81. We observe the same trend in LongBench (Table 2): when the gold evidence is surrounded by irrelevant context, PCED benefits from expert isolation.

Efficiency at Scale. Unlike context concatenation, which incurs high prefill costs, PCED leverages reusable per-document KV caches to reduce Time-To-First-Token (TTFT). As shown in Figure 3, with offline KV reuse PCED consistently achieves substantially lower TTFT across all top-K, with gains that scale to over $180\times$ faster TTFT (0.14s vs. 25.50s). Importantly, offline cache reuse is not required: when caches are built on the fly, PCED still delivers $2.5\times$–$6.6\times$ TTFT speedups as top-K increases from 8 to 64. This follows directly from the attention structure: concatenating N documents of length L induces prefill complexity $O((N \cdot L)^2)$, whereas PCED encodes documents independently at $O(N \cdot L^2)$ before applying decode-time aggregation. On long-context workloads (65k context tokens, 512 generated tokens), it also yields a $\sim$1.7$\times$ reduction in end-to-end latency. All results use a high-throughput setup with continuous batching and PagedAttention (Kwon et al., 2023) for both methods, validating the method's efficiency under realistic conditions.

Table 5: **Component Analysis.** Disentangling benefits of Contrastive Decoding vs. Retrieval Prior.

		Only Contrastive ($\gamma = 0$)	Only Retrieval ($\beta = 0$)	PCED
LLAMA-8B	HOTPOTQA	46	53	**64**
	NQ	52	70	**85**
MISTRAL-13B	HOTPOTQA	57	65	**66**
	NQ	71	80	**81**

Ablations. We verify that both terms in Eq. 2 are important: removing the retrieval prior (γ=0) or the contrastive calibration (β=0) leads to large accuracy drops (Table 5). We further find that Max aggregation best supports token-level expert switching in multi-hop QA, whereas soft mixtures can help in single-doc settings. Full sweeps over β, γ, aggregation rules, and top-k are in Appendix §C.

6 Conclusion

We presented PCED, a training-free decoding framework that adapts context-aware decoding to parallel, independently cached document experts. By combining expert-wise calibration against a no-context amateur, retrieval-prior gating, and token-level expert switching, PCED recovers cross-document reasoning without shared attention. Empirically, it matches or surpasses full-context baselines and is more robust to distractors. This offers an exciting alternative to long context models, allowing the number of documents to scale flexibly with batch size rather than being limited by the training context window.

Limitations

Despite its strong empirical performance and efficiency benefits, PCED has several limitations.

Dependence on access to model logits. PCED relies on per-expert token-level logits to perform retrieval-aware contrastive decoding, explicitly calibrating contextual experts against the amateur (prior) expert at each decoding step. This requirement assumes full access to the model's output logits. As a result, PCED cannot be directly applied to closed-source or API-only language models that expose only sampled tokens or log-probabilities for a limited subset of candidates. While this constraint is shared by many contrastive and guidance-based decoding methods, it currently restricts the applicability of PCED to open or self-hosted models.

Sensitivity to retrieval quality. Like most RAG approaches, PCED depends on the quality of the retrieved documents and their associated relevance scores. If relevant evidence is not retrieved or is assigned low relevance, the corresponding expert may be underweighted or never selected during decoding. Although retrieval-aware contrastive decoding mitigates noise from weak or irrelevant documents, it cannot recover evidence that is entirely absent from the candidate set. While our experiments use a single retrieval round, PCED is compatible with iterative or search-interleaved pipelines, as it modifies only how retrieved evidence is consumed. That said, in longer iterative settings, fixed retrieval and reranker scores may become stale as the generated reasoning trajectory evolves. Recent work shows that passage utility is not always well captured by static relevance alone (Perez-Beltrachini and Lapata, 2025). Extending PCED with step-wise score refresh or utility-based expert reweighting is an important direction for future work. A complementary future direction is to reduce reliance on external retrieval and reranking models altogether by explicitly training language models to accept parallel contextual inputs and learn end-to-end which expert to prioritize at each token, preserving parallelization at inference time.

Storage-Computation Trade-offs. PCED accelerates inference by effectively offloading online computation to offline storage. By persisting precomputed KV caches, the framework eliminates runtime encoding latency; however, this imposes a storage footprint that scales linearly with both corpus size and hidden state dimensionality. For instance, storing FP16 KV caches for the LOFT HOTPOTQA corpus (1,222 passages of 74 tokens on average) using LLAMA-3.1-8B necessitates approximately 11.04 GB of storage. As noted in §5, precomputed caches are optional: PCED retains significant speedups when encoding documents on the fly, making it viable also for dynamic corpora. When offline caching is used, however, PCED is optimally deployed in **read-heavy, write-rare** settings involving static corpora—such as enterprise knowledge bases—where the amortized storage cost is justified by the substantial reduction in query-time latency.

Acknowledgments

We thank the reviewers and the meta-reviewers for their comments, which helped us improve this work. We also thank Camilla Ferrario for helping to design the main figure of this paper. This work has been partially supported by the French government, through the 3IA Côte d'Azur Investments in the IA-cluster project managed by the National Research Agency (ANR) with the reference number ANR-23-IACL-0001.

References

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Brian J. Chan, Chao-Ting Chen, Jui-Hung Cheng, and Hen-Hsen Huang. 2025. Don't do rag: When cache-augmented generation is all you need for knowledge tasks. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 893–897, New York, NY, USA. Association for Computing Machinery.

Yihua Cheng, Yuhan Liu, Jiayi Yao, Yuwei An, Xiaokun Chen, Shaoting Feng, Yuyang Huang, Samuel Shen, Kuntai Du, and Junchen Jiang. 2025. Lmcache: An efficient kv cache layer for enterprise-scale llm inference. *arXiv preprint arXiv:2510.09665*.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on rag meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 6491–6501, New York, NY, USA. Association for Computing Machinery.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1).

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jonathan Ho and Tim Salimans. 2021. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Shufan Liu, Xuanzhe Liu, and Xin Jin. 2025. Ragcache: Efficient knowledge caching for retrieval-augmented generation. *ACM Trans. Comput. Syst.*, 44(1).

Jing Jin, Houfeng Wang, Hao Zhang, Xiaoguang Li, and Zhijiang Guo. 2024. DVD: Dynamic contrastive decoding for knowledge amplification in multi-document question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4624–4637, Miami, Florida, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien M. R. Arnold, Vincent Perot, Siddharth Dalmia, Hexiang Hu, Xudong Lin, Panupong Pasupat, Aida Amini, Jeremy R. Cole, Sebastian Riedel, Iftekhar Naim, Ming-Wei Chang, and Kelvin Guu. 2024. Can long-context language models subsume retrieval, rag, sql, and more? *ArXiv*, abs/2406.13121.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Songshuo Lu, Hua Wang, Yutian Rong, Zhi Chen, and Yaohua Tang. 2025. TurboRAG: Accelerating retrieval-augmented generation with precomputed KV caches for chunked text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6599–6612, Suzhou, China. Association for Computational Linguistics.

Dongyang Ma, Yan Wang, and Tian Lan. 2025. Block-attention for efficient prefilling. In *The Thirteenth International Conference on Learning Representations*.

Mistral AI. 2024. Mistral NeMo.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. YaRN: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations*.

Laura Perez-Beltrachini and Mirella Lapata. 2025. Uncertainty quantification in retrieval augmented question answering. *Transactions on Machine Learning Research*.

Zexuan Qiu, Zijing Ou, Bin Wu, Jingjing Li, Aiwei Liu, and Irwin King. 2025. Entropy-based decoding for retrieval-augmented large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4616–4627, Albuquerque, New Mexico. Association for Computational Linguistics.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. AdaCAD: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11636–11652, Albuquerque, New Mexico. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. 2025b. KVLink: Accelerating large language models via efficient KV cache reuse. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xinyu Yang, Tianqi Chen, and Beidi Chen. 2025c. APE: Faster and longer context-augmented generation via adaptive parallel encoding. In *The Thirteenth International Conference on Learning Representations*.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. 2025. Cacheblend: Fast large language model serving for rag with cached knowledge fusion. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 94–109, New York, NY, USA. Association for Computing Machinery.

Howard Yen, Tianyu Gao, and Danqi Chen. 2024. Long-context language modeling with parallel context encoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2588–2610, Bangkok, Thailand. Association for Computational Linguistics.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. 2024. Distserve: disaggregating prefill and decoding for goodput-optimized large language model serving. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation*, OSDI'24, USA. USENIX Association.

Zihan Zhou, Chong Li, Xinyi Chen, Shuo Wang, Yu Chao, Zhili Li, Haoyu Wang, Qi Shi, Zhixing Tan, Xu Han, Xiaodong Shi, Zhiyuan Liu, and Maosong Sun. 2025. LLM×MapReduce: Simplified long-sequence processing using large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27664–27678, Vienna, Austria. Association for Computational Linguistics.

A Evaluation Setup

This appendix details the prompt templates and instantiation protocols for each dataset. To ensure a fair comparison **across all methods** (Concatenation, KV-merge, MAPREDUCE, and PCED), we fix the underlying dataset fields, system prompt, context template, question template, and answer prefix, and vary *only* the mechanism of context incorporation. All experiments were executed with a fixed random seed (42) to ensure deterministic results. Unless otherwise stated, all reported numbers correspond to a single deterministic run per method.

Prompt Definitions. Each dataset instance is composed of three standardized fields: a `system_prompt` containing high-level instructions; a `context_template` which wraps the retrieved text; and a `question_template` applied to the user query.

A.1 LOFT Benchmark

We utilize the LOFT benchmark (Lee et al., 2024). Dataset statistics (e.g., number of examples, context lengths, and task distributions) are reported in Table 1 of the original LOFT paper (Lee et al., 2024).

A.1.1 LOFT-RAG Templates

For RAG tasks, all methods utilize the prompt configuration defined in Figure 4. The `{context}` slot is populated according to the specific method.

System Prompt You will be given a list of documents. You need to read carefully and understand all of them. Then you will be given a query, and your goal is to answer the query based on the documents you have read.
Context Template `{context}`
Question Template Based on the documents above, can you answer the following query? Write a concise answer. query: `{question}`

Figure 4: Prompt template configuration for LOFT-RAG tasks.

A.1.2 LOFT-ICL Templates

For In-Context Learning (ICL) tasks, we enforce a strict output format to facilitate automated parsing. The templates are defined in Figure 5.

System Prompt Please answer the following questions and ensure you follow a consistent format. In particular, ensure your final answer always looks like 'Output: ['your_answer_here']' After Output write ONLY the best option following the example(s). Do NOT write anything else.
Context Template Example(s): `{context}`
Question Template Now begin! `{question}`

Figure 5: Prompt template configuration for LOFT-ICL tasks.

A.2 Method-Specific Instantiations

PCED. PCED treats retrieved documents (RAG) or exemplars (ICL) as independent contextual experts. Concretely, for each query we create N contextual expert inputs by applying the dataset `system_prompt` and `context_template` to documents, yielding N separate (system, context) prompt instances. At decoding time, each expert produces logits conditioned on its own KV cache. We additionally include an **amateur** expert that represents the model prior: it is instantiated using `system_prompt` only. All experts share the identical `question_template`.

MAPREDUCE. This method involves a two-stage process. First, the *map* stage summarizes individual documents using the fixed instruction: *"Summarize the given documents concisely, focusing on the key points and main ideas."*

The resulting summaries are concatenated into a single prompt. In the subsequent *reduce* stage, the standard dataset templates (Figure 4 or 5) are used, with the concatenated summaries substituting the raw documents in the `{context}` slot.

A.3 LongBench

For LongBench (Bai et al., 2024), we strictly adhere to the official task instructions and question templates outlined in the original paper's Appendix B. Dataset statistics (e.g., number of examples, context lengths, and task distributions) are reported in Table 1 of the original LongBench paper (Bai et al., 2024).

A.4 Synthetic Dataset

To benchmark TTFT and end-to-end latency (Figure 3) under controlled context length, we construct a small synthetic dataset with fixed formatting and token budgets. Each instance contains $N{=}64$ documents; exactly one *gold* document includes a "secret code" string, while the remaining documents are distractors. We enforce an exact document length of 2048 tokens via padding/truncation. The query asks the model to output the secret code verbatim. We include a warmup sample to eliminate one-time initialization effects and stabilize latency measurements.

B Normalization of Retrieval and Reranker Scores

Motivation. PCED uses retrieval and reranker scores as a document-level prior (Eq. 2), where the prior enters as $\log r_k$. We therefore map all relevance signals to a common range $r_k \in [0, 1)$ (and clip away from 0 to avoid $\log 0$).

Retrieval scores (BGE-M3). Let s_k denote the raw retrieval score produced by `bge-m3` for expert k under a given scoring mode. Different modes have different score ranges, so we normalize as follows:

Dense / ColBERT. For the `dense` and `colbert` modes, similarity scores are bounded in $[-1, 1]$. We apply an affine rescaling:

$$r_k^{\text{ret}} = \text{clip}\left(\frac{s_k + 1}{2},\ 0,\ 1 - \epsilon\right), \tag{4}$$

which maps $[-1, 1] \mapsto [0, 1]$, followed by clipping to $[0, 1 - \epsilon)$.

Sparse. For the `sparse` mode, scores are non-negative and unbounded. Following standard practice for normalizing unbounded similarity/distance values into $[0, 1]$ (e.g., arctan-based normalization used in hybrid reranking), we apply a saturating arctan transform:

$$r_k^{\text{ret}} = \text{clip}\left(\frac{2}{\pi}\arctan(\max(s_k, 0)),\ 0,\ 1 - \epsilon\right). \tag{5}$$

This preserves monotonicity while smoothly compressing large sparse scores.

Reranker scores (BGE reranker). We use `BAAI/bge-reranker-v2-m3` via `FlagReranker`. With `normalize=True`, the reranker applies a sigmoid to map raw logits to $[0, 1]$:

$$r_k^{\text{rer}} = \sigma(z_k) = \frac{1}{1 + \exp(-z_k)}. \tag{6}$$

As above, we clip to $[0, 1 - \epsilon)$ before using the values in $\log r_k$.

Score fusion. After normalization, we combine retrieval and reranker signals into a single relevance score using the harmonic mean:

$$r_k = \frac{2\, r_k^{\text{ret}}\, r_k^{\text{rer}}}{r_k^{\text{ret}} + r_k^{\text{rer}} + \epsilon}. \tag{7}$$

In all experiments we set $\epsilon = 10^{-8}$.

C Ablations

In this section, we provide a detailed analysis of the hyperparameters governing PCED. Unless otherwise stated, all ablations are performed using the

PCED-Dense variant on the HOTPOTQA and Natural Questions (NQ) datasets, using both LLAMA-3.1-8B-INSTRUCT and MISTRAL-NEMO-13B-INSTRUCT.

C.1 Impact of Contrastive Strength (β)

The contrastive strength parameter β determines how aggressively the expert distribution (s_k) is sharpened against the amateur prior (s_0). We compare our default dynamic β strategy (derived from AdaCAD) against fixed values $\beta \in \{0.25, 0.5, 0.75, 1.0\}$. Additionally, we evaluate the setting $\beta = 0$, which effectively removes the contrastive component and relies solely on the retrieval prior and raw expert logits.

Results are presented in Table 6. We observe three key trends:

1. **Necessity of Contrastive Decoding ($\beta > 0$):** Setting $\beta = 0$ generally degrades performance compared to the best contrastive settings, confirming that subtracting the amateur logit helps isolate the specific knowledge provided by the retrieved document.

2. **Instability of Fixed β:** While specific fixed values can achieve high scores on individual tasks (e.g., $\beta = 0.25$ on LLAMA-NQ or $\beta = 0.75$ on LLAMA-HOTPOTQA), they are inconsistent. A value that works well for one dataset may fail on another (e.g., $\beta = 0.75$ drops significantly on MISTRAL-HOTPOTQA compared to lower values).

3. **Robustness of Dynamic β:** The dynamic strategy consistently delivers competitive performance across all models and datasets without requiring per-task tuning. We therefore select **Dynamic** as the default to ensure stability across diverse retrieval scenarios.

Table 6: **Ablation of Contrastive Strength (β).** We compare fixed β values against our Dynamic strategy. The column $\beta = 0$ represents standard decoding without the contrastive penalty (only retrieval prior). **Bold** denotes the best result.

		No CD	Fixed β				Ours
Model	**Dataset**	$\beta = 0$	0.25	0.50	0.75	1.0	**Dynamic**
LLAMA-8B	HOTPOTQA	53	65	61	**67**	59	64
	NQ	70	**88**	65	84	62	85
MISTRAL-13B	HOTPOTQA	65	62	62	58	54	**66**
	NQ	80	**83**	82	78	80	81

C.2 Sensitivity to Retrieval Prior (γ)

The parameter γ controls the influence of the retrieval/reranker scores on expert selection via the term $\gamma \log r_k$. We perform a sweep over $\gamma \in \{0.5, 1.0, 1.5, 2.0, 3.0, 4.0\}$ and compare these with our chosen default $\gamma = 2.5$.

Results are shown in Table 7. We observe the following:

- **Under-weighting ($\gamma < 1.5$):** Lower values often degrade performance (e.g., LLAMA-NQ drops significantly to 75 at $\gamma = 0.5$). This confirms that expert selection cannot rely on internal perplexity alone; strong external relevance signals are necessary to suppress distractors.

- **Over-weighting ($\gamma \geq 4.0$):** High values yield inconsistent results. While LLAMA-NQ peaks at $\gamma = 4.0$ (87), LLAMA-HOTPOTQA degrades compared to lower values (64 vs 66). Excessive gating forces the model to rigidly follow the retriever's ranking, potentially overriding valid reasoning from lower-ranked experts on complex queries.

- $\gamma = 2.5$**:** The range $\gamma \in [2.0, 3.0]$ represents a stable "sweet spot" across both models and datasets. We select $\gamma = 2.5$ as the default because it offers the best trade-off: it maximizes performance on difficult tasks like NQ (matching the high scores of $\gamma = 2.0 - 3.0$) while avoiding the instability seen at the extremes.

Table 7: **Sensitivity sweep for Retrieval Prior weight (γ).** We use Dynamic β for all runs. The main paper uses $\gamma = 2.5$.

		Gamma (γ)						Default
Model	**Dataset**	0.5	1.0	1.5	2.0	3.0	4.0	**2.5**
LLAMA-8B	HOTPOTQA	65	**66**	**66**	65	63	64	64
	NQ	75	84	86	85	85	**87**	85
MISTRAL-13B	HOTPOTQA	64	65	66	65	**67**	66	66
	NQ	78	79	80	**81**	**81**	**81**	81

C.3 Contrastive Signal vs. Retrieval Score Only

Finally, we isolate the contribution of the two core components of Equation 2: the contrastive signal (β) and the retrieval prior (γ).

Table 8 compares the full PCED method against two ablations:

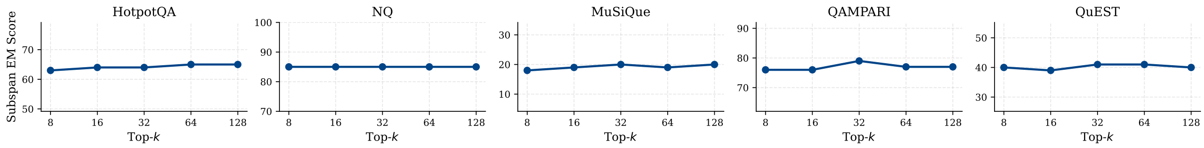

Figure 6: **Performance Stability across Top-k.** PCED maintains consistent accuracy from $k = 8$ to 128, confirming that the retrieval prior effectively suppresses noise from additional distractors.

1. **Only Retrieval Scores ($\beta = 0$):** Expert logits are boosted by retrieval scores but not calibrated against the amateur.
2. **Only Contrastive ($\gamma = 0$):** Expert logits are calibrated via contrastive decoding, but all experts are treated as equally likely (flat prior), ignoring retrieval ranking.

The results reveal two distinct findings:

- **Retrieval Prior is Foundational ($\gamma > 0$):** The "Only Contrastive" setting fails catastrophically across all benchmarks (e.g., LLAMA-NQ drops to 52). This confirms that without the external guidance of the retriever to gate irrelevant experts, the model is overwhelmed by noise from distractors.
- **Contrastive Signal is an Amplifier ($\beta > 0$):** The impact of the contrastive term is model-dependent. For LLAMA-3.1, it is critical: removing it ("Only Retrieval") causes a massive drop (e.g., NQ falls from 85 to 70), suggesting that LLAMA requires the amateur subtraction to suppress its own priors and hallucinations. Conversely, MISTRAL is more robust, achieving strong performance with retrieval scores alone, though the full PCED framework still secures the highest absolute scores in all cases.

Table 8: **Component Analysis.** We disentangle the benefits of the Contrastive Decoding signal versus the Retrieval Prior.

		Only Contrastive (No Prior, $\gamma = 0$)	Only Retrieval (No CD, $\beta = 0$)	Full PCED
LLAMA-8B	HOTPOTQA	46	53	**64**
	NQ	52	70	**85**
MISTRAL-13B	HOTPOTQA	57	65	**66**
	NQ	71	80	**81**

C.4 Ablation of Expert Aggregation Rule

PCED aggregates experts via a token-wise **Max** operation. We compare this against two probability-space alternatives: **Mixture-of-Experts** (MoE, weighted sum) and **Product-of-Experts** (PoE, weighted product), where weights are derived from retrieval scores.

Table 9 shows that **Max** aggregation is critical for multi-hop reasoning (HOTPOTQA), outperforming MoE by 8 points (64 vs. 56). We hypothesize that Max enables sharper *token-level expert switching*, allowing different documents to dominate different generation steps without their distributions needing to agree. Conversely, on single-document tasks like NQ, MoE performs slightly better (87 vs. 85), suggesting that soft averaging can be beneficial when evidence is concentrated in one expert and retrieval priors are accurate.

Table 9: **Aggregation Rule Ablation.** Comparison of PCED (Max) vs. probabilistic aggregation (MoE, PoE).

Aggregation	HOTPOTQA	NQ
Max (Ours)	**64**	85
Mixture (MoE)	56	**87**
Product (PoE)	46	85

C.5 Robustness to Candidate Pool Size (k)

We evaluate the stability of PCED (Dense, LLAMA-3.1-8B) as we scale the number of retrieved experts from $k = 8$ to $k = 128$. Results are visualized in Figure 6.

We observe two trends:

- **Noise Tolerance:** Performance remains nearly constant across all datasets despite a $16\times$ increase in experts. For instance, NQ scores stay flat at $\sim$85, while HOTPOTQA fluctuates only marginally (63–65). This confirms that the retrieval prior ($\gamma \log r_k$) effectively gates low-relevance experts, preventing distractor accumulation.
- **Recall without Penalty:** While low k is often sufficient, the lack of degradation at $k = 128$ allows users to maximize recall for difficult queries without sacrificing generation quality.