Learning to Compose Evidence with Counterfactual Continuation Values

Anonymous authors
Paper under double-blind review

Abstract

Retrieval-augmented question answering selects passages that are consumed jointly, yet common selectors score passages independently or optimize hand-designed set proxies. We study counterfactual continuation-value learning for bounded evidence selection. Given a small candidate pool and a frozen reader, we evaluate every feasible passage subset, cache its answer utility, and derive action targets from the best utility reachable after adding each passage. A compact selector learns these targets by regression and composes a context sequentially, with an explicit stop action. On HotpotQA, it improves over BGE Top-4 on two disjoint 64-question test sets across two training seeds. On a fresh 256-question sample, the budget-5 selector reaches 53.9 EM / 65.8 F1 versus 47.7 / 59.7 for BGE Top-4 and 50.8 / 62.7 for All-8. Without retraining, it transfers to two independent 2WikiMultiHopQA samples: 38.3 / 46.1 versus 31.6 / 40.3 on 256 questions, and 34.9 / 41.8 versus 30.3 / 35.3 on a disjoint 768-question sample. On a matched 768-question 2Wiki pool, a GRPO refinement reaches 35.29 EM / 43.11 F1, exceeding E5 Top-5 by 2.21 / 4.59 points at similar token cost. Candidate pools retain annotated supports, so these results evaluate composition conditional on support availability.

1 Introduction

Retrieval-augmented systems commonly rank candidate passages and truncate the list to fit a context budget. But a reader consumes the selected passages together: one passage may anchor an answer, another may complete a reasoning chain, and a third may distract the reader. The value of an item is therefore conditional on the evidence already selected.

Recent work makes this set-level view explicit. MORSE optimizes budgeted evidence packages using handcrafted relevance, coverage, pairwise relation, redundancy, and efficiency signals (Ye et al., 2026), while SETR frames retrieval-augmented generation as evidence-set selection rather than passage ranking (Lee et al., 2025). We ask a more specific question: can exact outcomes from a frozen downstream reader supervise a learned sequential selector?

The obstacle is credit assignment. A myopic marginal score $U(S \cup \{i\}) - U(S)$ may assign no credit to an evidence item that only becomes useful with a later item. Terminal-reward reinforcement learning can address this in principle, but sampled set rewards provide noisy, coarse supervision. We instead exploit a tractable training setting: for a small candidate pool, evaluate all subsets once with a frozen reader, then compute the best downstream utility still reachable after each possible action.

We call this target a counterfactual continuation value. For selected set S, candidate i, and maximum size B, it is the maximum reader utility among feasible final sets containing $S \cup \{i\}$. The stop target is the utility of the current set. Offline execution converts the frozen reader's finite-pool outcomes into state-action supervision, which a compact value model learns directly and uses to select one item at a time.

Our experiments provide three findings. First, on two HotpotQA holdouts, the learned selector improves on BGE Top-4 for both training seeds, with gains varying by sample. Second, a frozen HotpotQA selector exceeds BGE Top-4 on two independent 2WikiMulti-

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

HopQA samples. Third, same-pool comparisons favor our learned composer over an in-house MORSE-style proxy on a 768-question 2Wiki sample and a fresh 256-question HotpotQA sample. A matched target-choice ablation further measures how continuation supervision affects selection at a fixed five-passage budget.

1.1 Related work

Retrieval-augmented generation combines retrieval with a reader or generator, with early systems based on latent retrieval, dense passage retrieval, and multi-passage generation (Guu et al., 2020; Karpukhin et al., 2020; Lewis et al., 2020; Izacard & Grave, 2021). Multi-hop QA work studies path retrieval, evidence selection, and reasoning over benchmark-specific support structures (Feldman & El-Yaniv, 2019; Xiong et al., 2021; Groeneveld et al., 2020; Jiang & Bansal, 2019; Fang et al., 2020; Trivedi et al., 2022; Ferguson et al., 2022; Khalifa et al., 2023; Zhang et al., 2024; Yang et al., 2018; Ho et al., 2020). More recent RAG systems learn when and what to retrieve or formulate retrieval as set selection, including SETR, MORSE, DRO, Self-RAG, RE-RAG, REAR, EfficientRAG, and Adaptive-RAG (Lee et al., 2025; Ye et al., 2026; Shi et al., 2025; Asai et al., 2024; Kim & Lee, 2024; Wang et al., 2024; Zhuang et al., 2024; Jeong et al., 2024). Our focus is narrower: supervise a sequential composer from exhaustive outcomes of a fixed reader over a small candidate pool.

Our formulation is also related to classical methods for diverse or budgeted set selection, including maximal marginal relevance, determinantal point processes, and submodular selection and approximation (Carbonell & Goldstein, 1998; Kulesza & Taskar, 2012; Lin & Bilmes, 2011; Nemhauser & Wolsey, 1978). Shapley values provide a general language for allocating value among contributors, while Q-learning and policy-gradient methods provide standard value- and reward-based learning approaches (Shapley, 1953; Watkins & Dayan, 1992; Williams, 1992; Schulman et al., 2015; 2017). Decision-focused learning connects predictive models to downstream optimization objectives (Elmachtoub & Grigas, 2022); in contrast, our selector directly regresses on attainable reader utility for each state-action pair. We evaluate the learned selector against retrieval and subset-selection baselines and compare continuation and immediate-utility training targets directly.

2 Problem and method

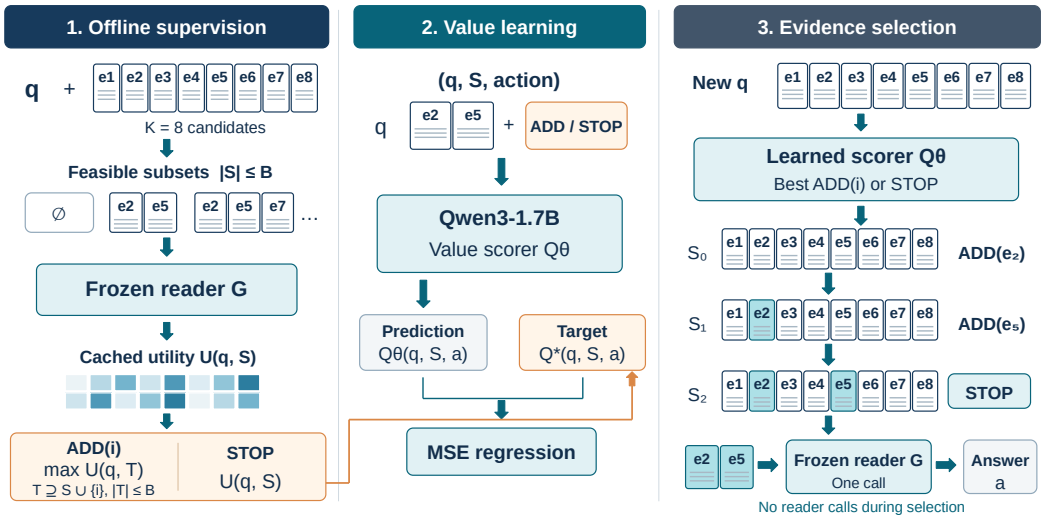

Figure 1: Continuation-value evidence composition. Left: a frozen reader evaluates subsets offline to derive exact ADD and STOP targets. Middle: a compact value scorer learns these targets by regression. Right: the learned scorer selects evidence sequentially, calling the reader only after selection. The selection trace and utility colors are schematic.

Figure 1 summarizes offline target construction, value learning, and sequential evidence selection.

2.1 Budgeted evidence composition

For a question q, let $C_q = \{e_1, \ldots, e_K\}$ be a candidate pool and let B be the maximum number of passages in the composed context. A frozen reader G answers using a subset $S \subseteq C_q$. We define task utility as

$$U(q, S) = \tfrac{1}{2}\,\mathrm{EM}(G(q, S)) + \tfrac{1}{2}\,\mathrm{F1}(G(q, S)) + \lambda_e\,\mathrm{SupportRecall}(S) - \lambda_c\,|S|/B. \tag{1}$$

In our main HotpotQA and 2Wiki runs, $\lambda_e = 0.15$ and $\lambda_c = 0.005$. EM and token-level F1 are computed against the annotated answer; support recall measures the fraction of annotated supporting passages selected. The reader is fixed while training the selector.

2.2 Exact counterfactual continuation targets

For any partial set S and legal action a, define

$$Q^*(q, S, a) = \begin{cases} \max\limits_{T:\, S\cup\{i\}\subseteq T,\ |T|\le B} U(q, T), & a = \mathrm{ADD}(i), \\ U(q, S), & a = \mathrm{STOP}. \end{cases} \tag{2}$$

The maximization ranges over subsets of the fixed candidate pool. We compute $U(q, T)$ for all 2^K subsets with the frozen reader once, cache the results, and derive every legal action target by table lookup and maximization. This target gives an item credit when it preserves a high-value completion, even if the current partial set receives no immediate reward. It also incorporates higher-order set effects without assuming that utility decomposes into singleton or pairwise terms.

Credit for complementary evidence. Let $V(q, S) = \max_{T\supseteq S,\,|T|\le B} U(q, T)$. A feasible completion either stops at S or adds an item before continuing, yielding

$$V(q, S) = \max\left\{U(q, S),\ \max_{i\in C_q\setminus S} V(q, S\cup\{i\})\right\}, \qquad |S| < B. \tag{3}$$

An ADD target therefore measures the quality of the best completion still available after committing to that item. An item can receive a high target even when its immediate answer utility is low, provided that it participates in a useful evidence combination. Conversely, an attractive singleton can have a lower target if selecting it consumes a slot needed by a better combination. With exact values, greedy ADD/STOP decisions preserve the best reachable utility until termination. This is the finite-horizon Bellman property of the subset table; Appendix B gives its derivation, an approximation-error bound, and a three-item example.

2.3 Learning and inference

The selector receives the question, the currently selected passages, and one proposed ADD or STOP action. A Qwen3-1.7B encoder with a scalar value head predicts $\widehat{Q}_\theta(q, S, a)$. We train a LoRA adapter and value head using mean squared error:

$$\mathcal{L}(\theta) = \mathbb{E}_{(q,S,a)}\left[\left(\widehat{Q}_\theta(q, S, a) - Q^*(q, S, a)\right)^2\right]. \tag{4}$$

At inference, the selector scores each legal action, takes the highest-valued action, and repeats until STOP or the budget is reached. The reader is called only to construct offline targets and to evaluate final selections; no reader query is needed during selection. We serialize a selected set in canonical candidate order so the state representation does not depend on the order in which passages were added.

Offline and online computation. For $K = 8$, the offline cache contains 256 reader outputs per training question, or 32,768 outputs over the 128-question supervision pool. The same cache supplies targets for different budgets and utility weights without rerunning the reader: EM, F1, support recall, and subset size are already recorded. At test time, a state of size $s < B$ requires $K - s$ ADD scores and one STOP score. A trajectory that fills its budget thus evaluates at most

$$\sum_{s=0}^{B-1}(K - s + 1) = B(K + 1) - \frac{B(B-1)}{2} \tag{5}$$

action prompts: 30 for $B = 4$ and 35 for $B = 5$. These are scalar predictions from the 1.7B selector, batched within each decision round. The 8B reader runs once on the final evidence set. Early stopping reduces the number of decision rounds and selected passages.

Relation to proxy subset optimization. MORSE searches subsets using observable candidate-level and pairwise proxies, then applies safeguards relative to an upstream ranking (Ye et al., 2026). Our signal instead comes from the frozen reader's actual answer outcome for each subset. DRO directly optimizes retrieval and generation through an answer-quality objective and a latent-document-order formulation (Shi et al., 2025). Our setting freezes the reader and uses exhaustive small-pool counterfactuals to produce supervised continuation targets.

3 Experimental setup

Training. We train on 128 HotpotQA training questions, each with $K = 8$ passages and all 256 candidate subsets evaluated by Qwen3-8B. The selector is Qwen3-1.7B with LoRA and a scalar value head. For the budget-4 selector, the training/validation split and target table are fixed; a second seed changes only model initialization. The budget-5 transfer model uses the corresponding budget-5 targets and training configuration.

Supervision and optimization details. The budget-5 run uses 116 training questions and 12 validation questions. We retain every empty and singleton state, then sample eight states at each size from two through five using a fixed question-specific seed. Each retained state contributes its STOP target and every legal ADD target. This gives 41 states and 225 action labels per question: 26,100 training labels and 2,700 validation labels. Labels are exact with respect to the cached subset outcomes even though training states are sampled. We train for two epochs with LoRA rank 16, scaling factor 32, dropout 0.05, learning rate 2×10^{-4}, batch size four, and eight-step gradient accumulation. Inputs are capped at 2,048 tokens. The scalar head reads the final token representation. Checkpoint selection minimizes mean validation action regret, breaking ties by validation MSE; the selected budget-5 checkpoint is from epoch one.

Policy refinement. The separately labeled GRPO-style variant uses an expanded HotpotQA pool of 244 training and 12 validation questions. Starting from the supervised selector, it samples four trajectories per question at temperature 0.35 and applies policy-gradient updates with within-group standardized terminal utility. We use two iterations and learning rate 10^{-6}, selecting the iteration by validation utility. The reader remains frozen.

Evaluation. HotpotQA uses two 64-question held-out sets disjoint from the selector's training questions; both sets are evaluated with two training seeds. We report the frozen selector against BGE Top-4, using the same Qwen3-8B reader and canonical answer order. For cross-dataset transfer, we evaluate the frozen HotpotQA selector on two independently sampled 2WikiMultiHopQA sets ($n = 256$ and $n = 768$). Each 2Wiki candidate pool contains annotated support passages by construction, with distractors added to make $K = 8$; the evaluation therefore isolates composition conditional on support availability.

Baselines and budgets. HotpotQA comparisons include BGE Top-2, BGE Top-4, and All-8. On the 768-question 2Wiki pool, we additionally compare fixed Top-4 and Top-5 rankings from BM25, BGE, Contriever, and E5 (Robertson & Zaragoza, 2009; Xiao et al., 2023; Izacard et al., 2022; Wang et al., 2022). The reported 2Wiki Composer uses budget five; retriever Top-5 baselines select five passages. We report mean passage count and reader input tokens. All-8 is a context-coverage reference.

3.1 Dataset and evaluation-sample statistics

Table 1 summarizes the benchmarks' evidence structure, and Table 2 reports the samples used in our experiments. Figures 2 and 3 show evaluation-set size and annotated support-count composition. HotpotQA contributes 384 held-out questions across two 64-question sets and a fresh 256-question sample; all contain exactly two annotated supports. The 1,024-question 2Wiki transfer set contains questions with either two or four supports, with four supports in 22.7% of cases. The fixed pool size is $K = 8$ in both datasets, while support cardinality differs. The selector is trained on 128 HotpotQA questions and evaluated on 2Wiki without 2Wiki training.

Table 1: Evidence structure of the benchmarks in our evaluation.

Dataset	Benchmark evidence structure	Role in this study
HotpotQA	Multi-hop questions with annotated supporting facts over Wikipedia passages (Yang et al., 2018).	Selector training and three held-out evaluations; support evidence is retained in each $K = 8$ candidate pool.
2WikiMultiHopQA	Multi-hop questions with annotated reasoning paths and supporting evidence, including bridge and comparison reasoning (Ho et al., 2020).	Two disjoint zero-shot transfer evaluations using the frozen HotpotQA selector; no 2Wiki selector training.

Table 2: Dataset statistics for the experiments reported in this paper. Counts describe our selected experimental samples, not the full benchmark releases.

Dataset	Selector training	Evaluation sample	Protocol, pool size, and annotated support passages
HotpotQA	128 questions	$2 \times 64 + 256$ (384 total; disjoint)	Held-out evaluation; $K = 8$; exactly 2 supports per question (384/384; 100%).
2WikiMultiHopQA	0 questions	256 + 768 (1,024 total; disjoint)	Zero-shot with frozen HotpotQA selector; $K = 8$; 2 supports: 792 (77.3%), 4 supports: 232 (22.7%).

Candidate pools retain annotated supports and are padded with distractors. On 2Wiki, the Composer uses budget 5 and BGE uses Top-4. Dataset descriptions: HotpotQA (Yang et al., 2018); 2WikiMultiHopQA (Ho et al., 2020).

4 Results

4.1 Held-out HotpotQA

Table 3 reports the budget-4 selector on the two held-out sets. Both training seeds improve EM over BGE Top-4 in both samples. The magnitude is not uniform: on the second holdout, seed 42 is close to the baseline in F1 while seed 43 has a larger gain. On the first holdout, the seed-42 Composer uses 459.9 selected tokens on average, compared with 484.8 for BGE Top-4 and 1051.3 for All-8.

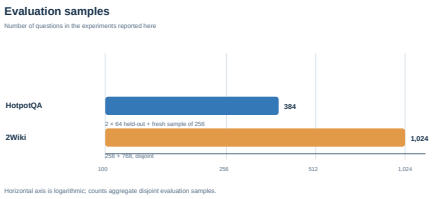

Figure 2: Evaluation sample sizes. Samples are disjoint within each benchmark; the horizontal axis is logarithmic.

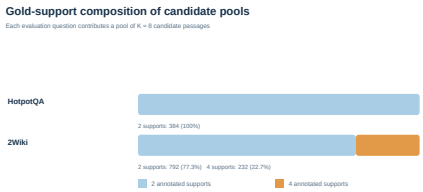

Figure 3: Annotated support-passage counts. All candidate pools contain eight passages and retain the supports.

Table 3: HotpotQA holdouts ($n = 64$ each), EM / F1 (%).

Test set	Method	Seed 42	Seed 43
A	BGE Top-4	46.9 / 56.4	
A	Continuation-value Composer	53.1 / 64.2	53.1 / 63.6
B	BGE Top-4	48.4 / 64.2	
B	Continuation-value Composer	50.0 / 64.3	53.1 / 68.8

4.2 Fixed-pool HotpotQA baselines

On a fresh 256-question HotpotQA pool, the supervised budget-5 Composer improves over BGE Top-4, BGE Top-2, and All-8 (Table 4). With training seed 42, it reaches 53.9 EM / 65.8 F1. Its EM gain over BGE Top-4 is 6.25 points, with a paired 95% bootstrap interval of [1.56, 10.94]. It uses 646 selected-passage tokens on average, compared with 494 for BGE Top-4 and 1,083 for All-8.

Table 4: Fixed-pool HotpotQA evaluation on 256 questions. The same Qwen3-8B reader scores every selection.

Selector	EM	F1	Mean passages	Mean tokens
BGE Top-2	43.8	53.9	2.00	226.8
BGE Top-4	47.7	59.7	4.00	494.1
All-8	50.8	62.7	8.00	1083.3
Composer, budget 4 (seed 42)	49.2	61.2	3.92	468.8
Composer, budget 5 (seed 42)	53.9	65.8	5.00	646.1

4.3 Evidence budget and answer quality

The fresh HotpotQA sample also compares budget-4 and budget-5 versions trained from the same 128 questions and cached reader outcomes (Table 4). Raising the budget from four to five increases EM by 4.69 points and F1 by 4.60 points. The corresponding contexts grow from 469 to 646 selected-passage tokens. With budget four, Composer improves over BGE Top-4 by 1.56 EM and 1.50 F1 points while using 5.1% fewer tokens. With budget five, it exceeds All-8 by 3.13 EM and 3.12 F1 points while using 40.4% fewer tokens.

These comparisons expose two useful operating points. Budget four offers a smaller context than BGE Top-4 on average, whereas budget five provides higher answer quality on this sample. Providing all eight candidates does not maximize accuracy: the learned five-passage context scores higher despite containing fewer passages. The result supports selecting a useful evidence combination instead of treating additional context as automatically beneficial. Token counts measure selected passage text; they exclude the selector's own computation.

4.4 Zero-shot transfer to 2WikiMultiHopQA

Table 5 and Figure 5 report two disjoint cross-dataset evaluations with the HotpotQA-trained Composer frozen. On the 256-question sample, it improves EM by 6.64 points and F1 by 5.75 points over BGE Top-4; paired bootstrap 95% intervals are [1.17, 11.72] for EM and [0.49, 11.00] for F1. On the independent 768-question sample, the gains are 4.56 EM points and 6.50 F1 points. The 768-question sample gives a larger estimate of transfer performance, with the same positive direction for both metrics. The Composer selects five passages on average (587 tokens), while BGE Top-4 uses four (481 tokens).

Table 5: Zero-shot 2WikiMultiHopQA transfer. Composer uses budget 5; BGE baseline uses Top-4. Values are EM / F1 (%).

Independent sample	Method	EM	F1
$n = 256$	BGE Top-4	31.6	40.3
$n = 256$	Continuation-value Composer	38.3	46.1
$n = 768$	BGE Top-4	30.34	35.34
$n = 768$	Continuation-value Composer	34.90	41.84

4.5 Matched-pool comparisons with external selectors

We compared the GRPO-updated composer against four fixed-pool retrievers and an in-house MORSE-style proxy using the same Qwen3-8B reader on 2Wiki ($n = 768$; Table 6). E5 Top-5 is the strongest retriever baseline at 33.07 EM / 38.52 F1. The composer reaches 35.29 / 43.11, with paired 95% bootstrap intervals for its gains of [-0.26, 4.69] EM and [2.06, 7.18] F1 points; mean input lengths are 583 and 572 tokens. Against the MORSE-style proxy, gains are 3.52 EM and 5.64 F1 points, with intervals [0.91, 6.25] and [2.95, 8.41]. On a fresh 256-question HotpotQA sample, the supervised budget-5 composer gains 3.91 EM and 5.01 F1 points over the proxy. Paired 95% intervals are [0.84, 9.36] for F1 and [-0.78, 8.59] for EM. The proxy comparison uses our implementation of the published objective.

Table 6: Fixed-pool comparisons on HotpotQA and 2WikiMultiHopQA. EM and F1 are percentages; tokens are mean reader-input tokens. 2Wiki retrievers use Top-5. Composer checkpoints differ by evaluation, as indicated.

Evaluation pool	Selector	EM / F1	Mean passages	Mean tokens
2Wiki, $n = 768$	GRPO Composer (iteration 2)	35.29 / 43.11	4.65	583.1
	E5 Top-5	33.07 / 38.52	5.00	571.6
	BGE Top-5	31.90 / 37.44	5.00	575.4
	Contriever Top-5	31.77 / 37.38	5.00	538.8
	BM25 Top-5	30.34 / 35.95	5.00	613.4
	MORSE-style proxy	31.77 / 37.47	5.00	581.6
HotpotQA, $n = 256$	Supervised Composer (budget 5)	53.91 / 65.81	5.00	646.1
	MORSE-style proxy	50.00 / 60.80	5.00	629.2
	BGE Top-5	50.00 / 60.56	5.00	626.8

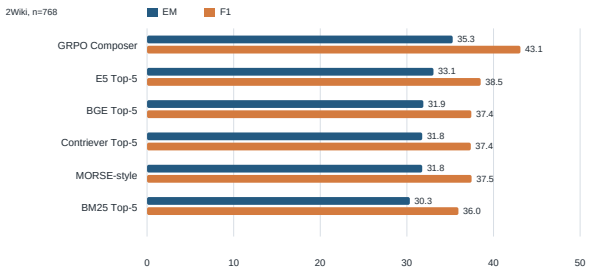

Figure 4: Matched 2WikiMultiHopQA comparison against four Top-5 retrievers and the MORSE-style proxy ($n = 768$). All selectors use the same Qwen3-8B reader.

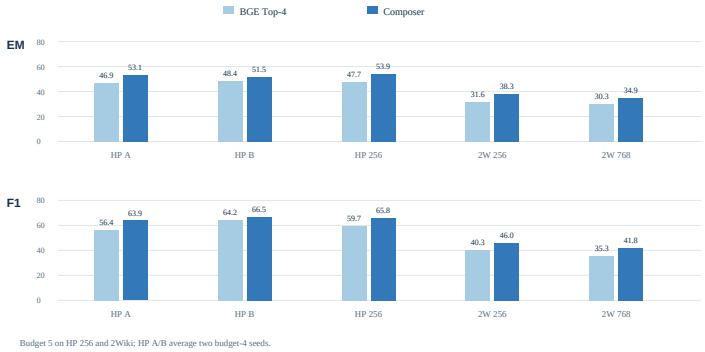

Figure 5: Answer accuracy on the evaluated samples. HotpotQA A/B scores average two selector seeds; the fresh 256-question HotpotQA score uses a supervised budget-5 checkpoint. On 2Wiki, transfer is zero-shot. The Composer uses budget 5 on the fresh HotpotQA and 2Wiki samples, versus BGE Top-4.

4.6 Continuation targets versus immediate utility

We isolate the target choice by matching the training questions, initialization seed, model configuration, and a fixed five-passage output. On HotpotQA, continuation improves EM by 1.95 points, while immediate utility obtains 0.43 higher F1. On the existing 768-question 2Wiki pool, we evaluate both matched checkpoints on one H200 and pool their reader prompts, so identical selected sets share the same generated answer. Continuation supervision improves the transfer point estimates by 1.04 EM and 1.08 F1 points (Table 7). The transfer gain is measured with identical output cardinality; continuation selects 589 tokens on average, compared with 578 for immediate utility. Table 7 reports the complete paired estimates for both datasets.

Table 7: Target-choice ablation with exactly five passages. Intervals resample paired questions; both checkpoints use training seed 42. The 2Wiki experiment prespecifies F1 as its primary metric.

Sample	Training target	EM	F1
HotpotQA, $n = 256$	Continuation	53.13	64.12
	Immediate utility	51.17	64.55
	Difference [95% CI]	+1.95 [-1.56, 5.47]	-0.43 [-3.85, 2.98]
2Wiki, $n = 768$	Continuation	35.16	42.47
	Immediate utility	34.11	41.39
	Difference [95% CI]	+1.04 [-1.43, 3.52]	+1.08 [-1.42, 3.59]

4.7 Policy refinement

We tested group-relative policy optimization and dense-reward PPO initialized from the supervised Composer. The expanded GRPO variant reaches 35.29 EM / 43.11 F1 on 2Wiki (Table 6), compared with 34.90 / 41.84 for the supervised selector. In the independent 768-question 2Wiki sample, the dense PPO checkpoint obtains 33.85 EM / 40.89 F1 versus 34.90 / 41.84 for supervised CUG. The observed effect therefore depends on the refinement recipe. We use the supervised selector for the primary results.

4.8 Question-level answer changes

We compare exact-match outcomes question by question on the fresh HotpotQA sample (Table 8). The budget-5 Composer corrects 27 questions that BGE Top-4 answers incorrectly, while losing 11 of its correct answers, for a net gain of 16 answers. Against All-8, the corresponding counts are 16 corrections and eight regressions. Thus, the aggregate gains reflect more corrected answers than lost answers against both a truncated ranking and the full candidate context. The shared successes and failures also account for every evaluated question, connecting the average EM scores to their underlying answer changes.

Table 8: Exact-match transitions on all 256 HotpotQA questions. "Composer only" counts corrected baseline errors; "baseline only" counts regressions.

Baseline	Both correct	Composer only	Baseline only	Both incorrect
BGE Top-4	111	27	11	107
All-8	122	16	8	110

5 Limitations and discussion

The MORSE comparison uses our implementation of its published proxy objective. Target-choice effects vary by dataset, and the fixed-budget ablations evaluate one training seed. Further seeds would characterize this source of variation.

The exhaustive target construction requires 2^K reader evaluations per question; our experiments use $K = 8$. The 2Wiki candidate pools retain annotated supports, so transfer evaluates composition after evidence has entered the candidate pool. The two primary HotpotQA holdouts contain 64 questions each, and their gains vary across samples and seeds. On transfer, the Composer uses a five-passage budget while BGE uses four.

6 Conclusion

We study counterfactual continuation-value supervision for sequential evidence composition. Exhaustive evaluations of a frozen reader provide action targets based on the best feasible completion after each selection. The learned selector improves answer quality over BGE Top-4 on HotpotQA and zero-shot 2Wiki samples. At budget five, it also outperforms the full eight-passage context on the fresh HotpotQA evaluation. Matched-pool comparisons with the GRPO-refined selector show gains over BM25, BGE, Contriever, E5, and a MORSE-style proxy. These results establish the usefulness of learned evidence composition in the evaluated candidate pools.

AI use statement

Generative AI tools were used to help polish the paper's writing and to support literature retrieval and discovery. The authors reviewed the AI-assisted edits and take responsibility for the final content.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–336, 1998. doi: 10.1145/290941.291025. URL https://www.cs.cmu.edu/afs/cs/Web/People/jgc/publication/MMR_DiversityBased_Reranking_SIGIR_1998.pdf.

Adam N. Elmachtoub and Paul Grigas. Smart predict, then optimize. Management Science, 68(1):9–26, 2022. doi: 10.1287/mnsc.2020.3922. URL https://doi.org/10.1287/mnsc.2020.3922.

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. Hierarchical graph network for multi-hop question answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 8823–8838, 2020. doi: 10.18653/v1/2020.emnlp-main.710. URL https://aclanthology.org/2020.emnlp-main.710/.

Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2296–2309, 2019. doi: 10.18653/v1/P19-1222. URL https://aclanthology.org/P19-1222/.

James Ferguson, Hannaneh Hajishirzi, Pradeep Dasigi, and Tushar Khot. Retrieval data augmentation informed by downstream question answering performance. In Proceedings of the Fifth Fact Extraction and VERification Workshop, pp. 1–5, 2022. doi: 10.18653/v1/2022.fever-1.1. URL https://aclanthology.org/2022.fever-1.1/.

Dirk Groeneveld, Tushar Khot, Mausam, and Ashish Sabharwal. A simple yet strong pipeline for hotpotqa. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020. URL https://aclanthology.org/2020.emnlp-main.711/.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval-augmented language model pre-training. In Proceedings of the 37th International Conference on Machine Learning, volume 119, pp. 3929–3938, 2020. URL https://arxiv.org/abs/2002.08909.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In Proceedings of the 28th International Conference on Computational Linguistics, pp. 6609–6625, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 874–880, 2021. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74/.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. Transactions on Machine Learning Research, 2022.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 7036–7050, 2024. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Yichen Jiang and Mohit Bansal. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop qa. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2726–2736, 2019. doi: 10.18653/v1/P19-1262. URL https://aclanthology.org/P19-1262/.

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 Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. Few-shot reranking for multi-hop qa via language model prompting. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, pp. 15882–15897, 2023. doi: 10.18653/v1/2023.acl-long.885. URL https://aclanthology.org/2023.acl-long.885/.

Kiseung Kim and Jay-Yoon Lee. Re-rag: Improving open-domain qa performance and interpretability with relevance estimator in retrieval-augmented generation. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 22149–22161, 2024. doi: 10.18653/v1/2024.emnlp-main.1236. URL https://aclanthology.org/2024.emnlp-main.1236/.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. Foundations and Trends in Machine Learning, 5(2–3):123–286, 2012. doi: 10.1561/2200000044. URL https://doi.org/10.1561/2200000044.

Dahyun Lee, Yongrae Jo, Haeju Park, and Moontae Lee. Shifting from ranking to set selection for retrieval augmented generation. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, pp. 17606–17619, 2025. doi: 10.18653/v1/2025.acl-long.861.

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. Retrieval-augmented generation for knowledge-intensive nlp tasks. In Advances in Neural Information Processing Systems, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 510–520, 2011. URL https://aclanthology.org/P11-1052/.

George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. Mathematics of Operations Research, 3(3):177–188, 1978. doi: 10.1287/moor.3.3.177. URL https://doi.org/10.1287/moor.3.3.177.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In Proceedings of the 32nd International Conference on Machine Learning, pp. 1889–1897, 2015. URL https://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Lloyd S. Shapley. A value for n-person games. In Contributions to the Theory of Games, volume 2, pp. 307–317. Princeton University Press, 1953. URL https://doi.org/10.1515/9781400881970-018.

Zhengliang Shi, Lingyong Yan, Weiwei Sun, Yue Feng, Pengjie Ren, Xinyu Ma, Shuaiqiang Wang, Dawei Yin, Maarten de Rijke, and Zhaochun Ren. Direct retrieval-augmented optimization: Synergizing knowledge selection and language models. arXiv preprint arXiv:2505.03075, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. Transactions of the Association for Computational Linguistics, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533, 2022.

Yuhao Wang, Ruiyang Ren, Junyi Li, Wayne Xin Zhao, Jing Liu, and Ji-Rong Wen. Rear: A relevance-aware retrieval-augmented framework for open-domain question answering. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 5613–5626, 2024. doi: 10.18653/v1/2024.emnlp-main.321. URL https://aclanthology.org/2024.emnlp-main.321/.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. Machine Learning, 8(3–4): 279–292, 1992. doi: 10.1007/BF00992698. URL https://doi.org/10.1007/BF00992698.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. arXiv preprint arXiv:2309.07597, 2023.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. Answering complex open-domain questions with multi-hop dense retrieval. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=EMHoBG0avc1.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2369–2380, 2018. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Qilei Ye, Yan Wang, and Chi Chen. Multi-objective evidence construction in retrieval-enhanced question answering: from ranking to subset optimization. Complex & Intelligent Systems, 2026. doi: 10.1007/s40747-026-02520-z.

Jiahao Zhang, Haiyang Zhang, Dongmei Zhang, Yong Liu, and Shen Huang. End-to-end beam retrieval for multi-hop question answering. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1718–1731, 2024. doi: 10.18653/v1/2024.naacl-long.96. URL https://aclanthology.org/2024.naacl-long.96/.

Ziyuan Zhuang, Zhiyang Zhang, Sitao Cheng, Fangkai Yang, Jia Liu, Shujian Huang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Efficientrag: Efficient retriever for multi-hop question answering. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pp. 3392–3411, 2024. doi: 10.18653/v1/2024.emnlp-main.199. URL https://aclanthology.org/2024.emnlp-main.199/.

A Deriving and computing continuation targets

This appendix expands the target construction in Section 2. The key object is a table of reader outcomes over a small, fixed candidate pool. It lets us ask a concrete counterfactual question before training: if the selector adds passage i now, what is the best utility it can still reach by filling the remaining slots?

A.1 Finite-pool value function

Fix a question q, candidate set C_q of size K, and budget B. For every subset $T \subseteq C_q$ with $|T| \leq B$, evaluate the same frozen reader and cache $U(q, T)$. Define the best reachable value from partial set S as

$$V_q(S) = \max_{T \in \mathcal{F}_B(S)} U(q, T), \qquad \mathcal{F}_B(S) = \{T : S \subseteq T \subseteq C_q,\ |T| \leq B\}. \tag{6}$$

For a legal add action, the target is exactly the best reachable value after taking that action; stopping keeps the current set:

$$Q^*(q, S, \mathrm{ADD}(i)) = V_q(S \cup \{i\}), \qquad Q^*(q, S, \mathrm{STOP}) = U(q, S). \tag{7}$$

The maximization includes the option to stop at any feasible completion. Thus an item is valuable when it preserves a good completion, even when the item alone does not improve the answer.

A.2 Target-construction procedure

With $K = 8$, subsets can be represented as K-bit masks. The reader is run once per subset, not once per action label. All ADD and STOP targets are then table lookups and maxima over cached values. The budget can be changed after caching without another reader call.

1. For each question, run the frozen reader on every candidate subset needed by the largest budget and store answer EM, token F1, support recall, and subset size.
2. Convert each cached subset outcome to $U(q, T)$ using Eq. 1.
3. For each training state S and legal passage i, take the maximum cached utility over all feasible supersets containing $S \cup \{i\}$; use $U(q, S)$ for STOP.
4. Regress the selector's scalar prediction for (q, S, a) on that exact target. At test time, score all legal ADD actions and STOP, choose the largest prediction, and repeat until STOP or the budget is reached.

For a pool of size K, exhaustive target construction requires at most 2^K reader evaluations per question (256 when $K = 8$). At a test state of size $s < B$, inference scores $K - s$ add actions and one stop action. A full B-addition trajectory therefore uses $B(K+1) - B(B-1)/2$ scalar scores: 35 when $K = 8, B = 5$. These are selector evaluations; the reader is used after selection to evaluate the final context.

B Properties of the exact target table

B.1 Bellman recursion and exact policy

For $|S| < B$, every feasible completion is either S itself or contains some legal next item i. Partitioning the feasible completion family in Eq. 6 gives

$$V_q(S) = \max\left\{U(q, S),\ \max_{i \in C_q \setminus S} V_q(S \cup \{i\})\right\}. \tag{8}$$

This is the finite-horizon Bellman recursion for the subset table, and Eq. 7 supplies exactly its action values.

Proposition 1 (exact values recover an optimal feasible set). If every decision selects an action maximizing the exact targets in Eq. 7, the resulting stop-enabled policy terminates at S_{final} with $U(q, S_{\text{final}}) = V_q(\varnothing)$.

Proof. At any state S with $|S| < B$, Eq. 8 says the largest ADD/STOP target equals $V_q(S)$. If the maximizing action is ADD(i), then $V_q(S \cup \{i\}) = V_q(S)$, so the next state retains the reachable optimum. If the maximizing action is STOP, its target is $U(q, S) = V_q(S)$ and termination is optimal. If the policy reaches size B, the only feasible completion is S itself, so $U(q, S) = V_q(S)$. Each ADD increases set size, hence the policy terminates after at most B additions. The argument does not require utility to be monotone, submodular, or additive. □

B.2 Approximate action values

The learned selector predicts $\widehat{Q}_\theta$ rather than the exact values. The following guarantee states what a uniform prediction-error bound would imply; it is not inferred from mean-squared error alone.

Proposition 2 (finite-horizon value loss). Suppose that at every state visited by the learned policy, every legal action satisfies $|\widehat{Q}_\theta(q, S, a) - Q^*(q, S, a)| \le \epsilon$. Then

$$U(q, S_{\text{final}}) \ge V_q(\varnothing) - 2B\epsilon. \tag{9}$$

Proof. Let a^* maximize the exact target at state S, and let $\widehat{a}$ maximize the predicted target. The two prediction errors and the maximizing property imply

$$\begin{aligned} Q^*(q, S, \widehat{a}) &\ge \widehat{Q}_\theta(q, S, \widehat{a}) - \epsilon && (10)\\ &\ge \widehat{Q}_\theta(q, S, a^*) - \epsilon && (11)\\ &\ge Q^*(q, S, a^*) - 2\epsilon = V_q(S) - 2\epsilon. && (12) \end{aligned}$$

If $\widehat{a}$ is ADD(i), its exact target is $V_q(S \cup \{i\})$, so the reachable value falls by at most 2ϵ. If it is STOP, the same inequality directly lower-bounds the terminal utility. A trajectory has at most B decisions that can lose reachable value: at most B ADD actions if it fills the budget, or at most $B - 1$ ADD actions followed by STOP otherwise. Summing the per-decision losses proves Eq. 9. □

The assumption is stronger than a small average validation MSE: it requires a bound for each legal action at states the policy actually visits. Our experiments do not establish this uniform condition, so Eq. 9 is a property of the target formulation, not a certified numerical guarantee for the trained selector. Forced-five evaluations also remove STOP and are controlled target-choice comparisons rather than evaluations of the stop-enabled policy.

C A worked complementarity example

Consider three passages $C = \{a, b, c\}$ and budget $B = 2$. Suppose the cached reader utilities are:

Set T	$U(T)$	Set T	$U(T)$
$\varnothing$	0.0	$\{a\}$	0.4
$\{b\}$	0.0	$\{c\}$	0.0
$\{a, b\}$	0.4	$\{a, c\}$	0.4
$\{b, c\}$	1.0		

Immediate utility ranks a first because $U(\{a\}) = 0.4$ while $U(\{b\}) = U(\{c\}) = 0$. It then cannot exceed 0.4 within the budget. Continuation values instead assign targets 0.4, 1.0, and 1.0 to ADD(a), ADD(b), and ADD(c) from the empty state. After choosing b, ADD(c) preserves value 1.0, so the policy reaches the complementary pair $\{b, c\}$. This example illustrates the credit-assignment difference; it is not an empirical result.

D Experimental protocol and paired target ablation

D.1 Training and evaluation details

Table 9 collects the implementation details used in the main experiments so that the target construction and evaluation can be reproduced without searching across sections.

Table 9: Reproduction details for the principal supervised selector.

Component	Setting
Candidate pool	$K = 8$ passages. Training pools use annotated supports plus distractors; the 2Wiki evaluation pools also retain annotated supports and therefore test composition conditional on support availability.
Offline reader	Frozen Qwen3-8B; all 256 subsets are evaluated for each training question. Utility is $0.5\,\mathrm{EM} + 0.5\,\mathrm{F1} + 0.15\,\mathrm{SupportRecall} - 0.005\lvert S\rvert/B$.
Selector	Qwen3-1.7B with a LoRA adapter (rank 16, scaling 32, dropout 0.05) and a scalar value head over the final token representation.
Training split	128 HotpotQA questions: 116 training and 12 validation. Eight states are sampled at each size 2 through 5; all empty/singleton states are retained. This yields 26,100 training and 2,700 validation action labels.
Optimization	Two epochs; AdamW learning rate 2×10^{-4}; batch size 4 with gradient accumulation 8; maximum input length 2,048 tokens. Select the checkpoint by validation action regret, breaking ties by MSE.
Inference	Canonical candidate order; greedy maximum predicted ADD/STOP value; stop at STOP or the passage budget. Reader generation is greedy with a 32-token answer limit in the reported cached-reader comparisons.
Uncertainty	Paired question bootstrap with 10,000 resamples and seed 2027 for the 768-question target-transfer comparison.

D.2 Continuation versus immediate-utility supervision

The target ablation changes only the ADD label: continuation uses the best feasible completion after adding a passage, while immediate utility uses the reader utility of the next subset. The STOP label, question/state/action examples, train/validation split, initialization seed, architecture, optimization settings, candidate pool, fixed five-passage output, reader, and canonical serialization are matched. Both selectors are scored on the same question; if their selected contexts are identical, the generated answer is shared. The reported estimates therefore compare the target choice under matched evaluation conditions.

On the 256-question HotpotQA sample, continuation has higher EM by 1.95 points while immediate utility has higher F1 by 0.43 points. On the independent 768-question 2Wiki pool, continuation is higher by 1.04 EM and 1.08 F1 points. The corresponding paired intervals are reported in Table 7. For the 2Wiki sample, continuation and immediate utility choose the identical evidence set on 113 questions; among the remaining questions, continuation wins 52 and loses 44 in EM, and wins 74 and loses 64 in F1 (zero-difference outcomes are ties). The point estimates favor continuation on this transfer sample, while variation across individual questions remains substantial. These fixed-budget results do not isolate an advantage in every metric or dataset, and the evaluated target ablation uses one training seed.

E Question-level illustrations

The cases below come from the paired 768-question 2Wiki transfer evaluation. They illustrate how changing the selected context can change the answer; they are individual examples, not evidence of average performance.

A case where continuation helps. For "Which film has the director who is older, Consider The Source (Web Series) or My Best Friend's Wife?", the gold answer is "My Best Friend's Wife". The continuation selector chose passages about both films and a passage identifying Vincenzo Salemme, and the frozen reader returned the correct film (EM/F1 = 1/1). The immediate-utility selector omitted the passage about Consider the Source and the reader named that web series instead (EM/F1 = 0/0). The selected context changes which side of the comparison is directly represented.

A case where continuation does not help. For "Who is Henry I of England's mother?", the gold answer is "Matilda of Flanders". The continuation-selected context included passages about William the Conqueror and Henry I, but the reader answered "Herleva" (EM/F1 = 0/0). The immediate-utility-selected context also lacked a direct Matilda passage, yet the reader returned "Matilda of Flanders" (EM/F1 = 1/1), consistent with answer knowledge not fully determined by the selected evidence. This case cautions against treating passage selection alone as a guarantee of answer correctness.

Together, the examples make the interface between selector and reader explicit: the selector changes the evidence supplied to the frozen reader, while the final answer can still reflect both supplied context and the reader's learned prior knowledge.

E.1 Paired target-ablation diagnostics

For completeness, Table 10 collects the paired effect sizes and context costs in one place. Both methods select exactly five passages. The 2Wiki answers were generated from 2,765 unique reader prompts after deduplicating identical selected contexts across the two policies.

Table 10: Additional paired diagnostics for the target ablation. Accuracy differences are continuation minus immediate utility, with paired 95% bootstrap intervals.

Sample	EM difference; 95% CI	F1 difference; 95% CI	Tokens C/I; same sets
HotpotQA	$+1.95$; $[-1.56, 5.47]$	-0.43; $[-3.85, 2.98]$	651.1 / 638.7; –
2Wiki	$+1.04$; $[-1.43, 3.52]$	$+1.08$; $[-1.42, 3.59]$	589.2 / 578.4; 113

Here C/I means continuation/immediate utility. The sample sizes are 256 HotpotQA and 768 2Wiki questions.

On 2Wiki, the per-question EM comparison has 52 continuation wins, 672 ties, and 44 losses; F1 has 74 wins, 630 ties, and 64 losses. The mean context-length increase is 10.8 tokens for continuation (paired 95% interval: 0.6 to 21.1). These diagnostics show that the answer-quality point gains are measured at the same passage count, with a small increase in average reader input length.