LONG-ZERO: LONG-CONTEXT POST-TRAINING VIA ADAPTIVE INFORMATION DEGRADATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective long-context post-training requires supervision that encourages models to use document evidence. We introduce Long-Zero, a framework that derives ordinal supervision from controlled evidence degradation. Given a document and question, Long-Zero constructs nested context views by progressively redacting query-relevant chunks and generates an answer from each view. Their source-view order supplies an automatically checkable target for a policy that ranks the shuffled answers using the complete document. A feedback controller adjusts the relevance gap between views, adapting task construction to the policy's training feedback. This procedure uses document–question pairs without reference answers in its ranking reward. Across two model scales and two decoding modes, Long-Zero improves long-context QA Overall scores over the corresponding base models by 0.8–2.8 points. It also exceeds SFT on 7B CoT QA Overall and achieves higher 4B CoT Hard and Long scores. On HELMET MS MARCO reranking, macro NDCG@10 improves over Base by 4.48 points for 7B and 1.72 for 4B, and over SFT by 5.45 and 2.05 points. Training trajectories show increasing ranking rewards alongside feedback-responsive curriculum adjustments. Together, these results demonstrate the potential of controlled evidence degradation as a source of useful supervision for long-context post-training.

1 INTRODUCTION

Long context windows do not by themselves ensure reliable use of evidence. Models must identify relevant passages, reconcile information distributed across a document, and distinguish supported answers from plausible alternatives. Evaluations of long-context models expose a gap between nominal window size and effective reasoning capacity (Hsieh et al., 2024; Bai et al., 2024b;c; Yen et al., 2025). Closing this gap requires post-training signals that reward evidence use, and distinguish reliable document-grounded reasoning from merely processing longer sequences.

Obtaining such signals is difficult for open-ended tasks. Supervised fine-tuning (SFT) relies on instruction–response demonstrations (Bai et al., 2024a); preference-based approaches use synthesized comparisons or AI feedback (Zhang et al., 2024; Tang et al., 2024). Reinforcement learning with verifiable rewards succeeds when an answer can be checked by a task-specific procedure, as in mathematics and code (Shao et al., 2024; Guo et al., 2025). Long-document synthesis rarely offers a comparable outcome verifier: several answers may be acceptable, and checking their claims can require interpreting and cross-referencing evidence throughout the original document.

We investigate the information available *before* an answer is generated as a source of supervision. For a document and question, we construct nested views by removing progressively more query-relevant content. A generator answers the same question under each view. The known source-view order supplies an ordinal target that can be checked without reference answers. Our modeling hypothesis is that access to more relevant evidence tends to improve answer quality. It connects an intervention on the input document to a training task over the generated answers, allowing the learner to practice comparisons grounded in the original evidence.

Long-Zero turns this hypothesis into a post-training procedure (Figure 1). A query-aware relevance scorer determines which chunks to redact. Repeated filler tokens preserve the length of the constructed token arrays. The policy receives the complete document and shuffled candidates, and is

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.

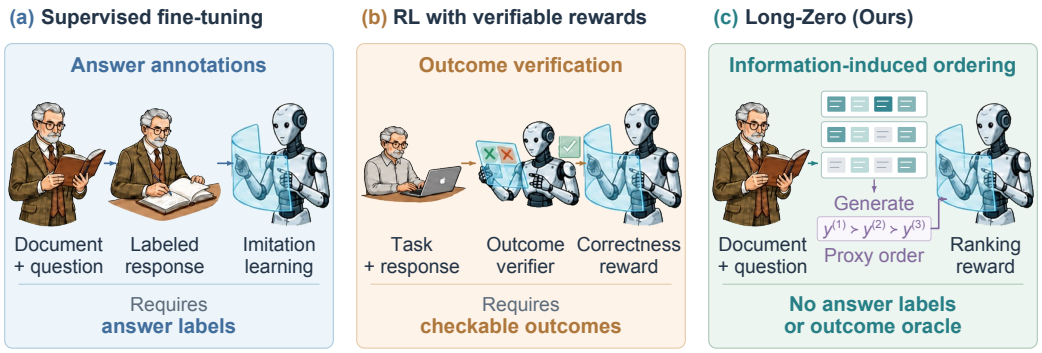

Figure 1: **Supervision from controlled evidence availability.** SFT uses target responses and conventional RLVR checks task outcomes. Long-Zero constructs a source-view order by degrading the evidence available during candidate generation. The learner ranks shuffled candidates under the full document. The intervention supplies an automatically checkable source-view target for document-grounded candidate comparison.

trained to recover the order induced by their source views. An adaptive controller adjusts the relevance mass removed between views: stronger feedback reduces the gap, while weaker feedback increases it. This curriculum changes the training tasks along an interpretable information axis.

Long-Zero improves over the corresponding base models in all four reported long-context QA model–decoding settings and at both scales on HELMET reranking. Relative to SFT, it achieves higher 7B CoT QA Overall and higher reranking macro scores at both scales. The 4B CoT results additionally show gains on the Hard and Long subsets, highlighting promising applications to demanding questions and extended contexts. Training trajectories complement these downstream comparisons: ranking rewards increase over training, and the controller adjusts the relevance gap in response to its feedback thresholds. These results connect the construction of information-derived supervision to measurable improvements in long-context tasks.

The contribution is threefold. First, we formulate information degradation as a source of answer-label-free ordinal supervision for long-context post-training. Second, we develop a framework that combines nested relevance-based redaction, full-context candidate comparison, and feedback-driven curriculum control. Its training target is constructed automatically from the source views and evaluated through a ranking reward. Third, we evaluate the framework on long-context QA and passage reranking, connecting downstream performance with observed training dynamics. Together, the formulation and experiments provide a concrete approach to using controlled evidence availability in post-training.

2 Long-Zero

Given document–question pairs $\mathcal{D} = \{(x_i, q_i)\}_{i=1}^{N}$, Long-Zero trains a policy using automatically constructed ranking targets. Its inputs are documents and questions, a pretrained relevance scorer, and pretrained language models. Figure 2 illustrates how evidence degradation, candidate generation, ranking, and feedback control fit together. The evaluated procedure uses the ranking reward defined in Section 2.3.

Learning signal and information access. The construction separates three roles. The generator answers a question from one context view; the policy compares answers against the original document; the verifier checks the predicted order against the source-view order without reading the answers. The intervention determines which evidence is available, while the generator determines how that evidence affects candidate quality. For example, a question requiring both a document's conclusion and its supporting rationale may yield a less complete answer when the rationale is removed. The complete document lets the policy inspect both the conclusion and supporting passages when comparing the generated answers.

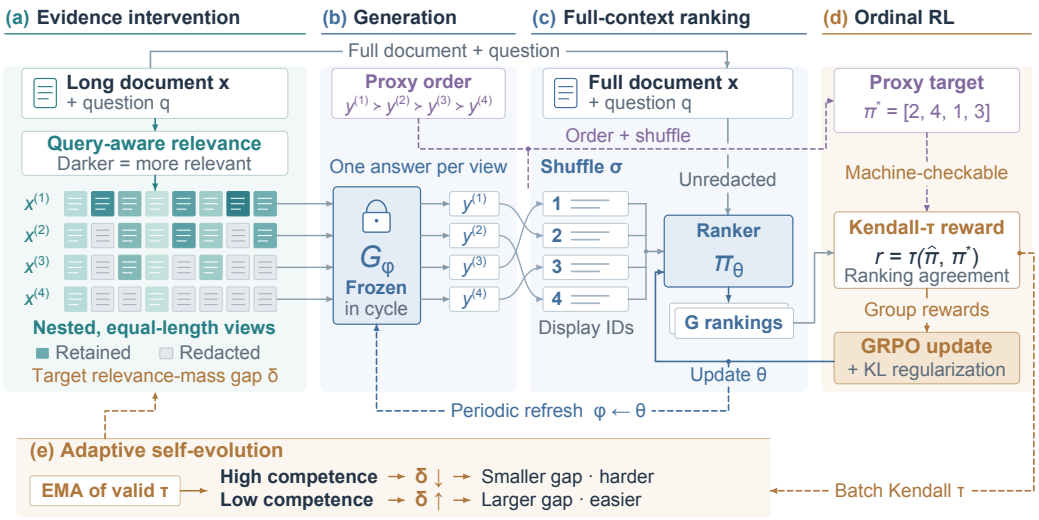

Figure 2: **Long-Zero framework.** Relevance-based redaction constructs nested source views. A generator answers each view, and the policy ranks shuffled candidates under the complete document. Source-view labels provide automatically checkable targets for ranking and controller feedback. The Kendall-τ panel depicts the candidate-pair formulation; the evaluated training procedure uses the ranking reward in Equation 7. The policy-to-generator arrow represents an optional refresh extension. The controller adapts the relevance gap used in subsequent view construction.

2.1 Constructing Information-Ordered Views

Partition the tokenized document x_i into M_i non-overlapping chunks $c_{i,1}, \ldots, c_{i,M_i}$ of size B, except possibly the last. A query-aware cross-encoder scores each chunk, using surrounding boundary context when available. Let $s_{i,j} \geq 0$ be its relevance score and define

$$w_{i,j} = \frac{s_{i,j}}{\sum_m s_{i,m}}, \qquad \sum_j w_{i,j} = 1. \tag{1}$$

An all-zero score vector falls back to uniform weights. Write ρ_i for the chunk indices sorted by descending weight. The normalized weights provide a common scale for selecting evidence within each document, while the sorted order specifies the sequence of chunk removal.

Why relevance mass and nested removal? Masking a fixed number of chunks creates different interventions when evidence is concentrated in a few passages versus spread throughout a document. Relevance-mass normalization supplies a document-relative control scale. Removing the highest-scoring chunks first targets content expected to matter for the question, whereas random removal may primarily discard irrelevant passages. Cumulative removal also gives successive views a common direction of change: later views cannot restore a passage removed earlier. Independent masks would instead yield incomparable evidence sets, making their source order harder to justify.

Starting with $\mathcal{M}_i^{(1)} = \varnothing$, we construct K views. At transition k, let $R_i^{(k)} = \sum_{j \notin \mathcal{M}_i^{(k)}} w_{i,j}$ be the remaining relevance mass. When $R_i^{(k)} \geq d_{i,t}$, a shared pointer selects the smallest prefix $\mathcal{S}_i^{(k+1)}$ of the remaining order ρ_i whose mass reaches the target gap $d_{i,t}$:

$$\sum_{j \in \mathcal{S}_i^{(k+1)}} w_{i,j} \geq d_{i,t}, \qquad \mathcal{M}_i^{(k+1)} = \mathcal{M}_i^{(k)} \cup \mathcal{S}_i^{(k+1)}. \tag{2}$$

If $R_i^{(k)} < d_{i,t}$, we instead select all remaining chunks; Equation 2's mass inequality applies only to the sufficient-mass branch. Once no chunks remain, subsequent transitions leave the masked set unchanged. Each transition operates on whole chunks, and the accumulated removed mass reaches or exceeds the target until the document is exhausted. This yields a nested sequence of retained evidence sets with an explicit source order.

Illustrative construction. Let the sorted chunk weights be 0.34, 0.26, 0.18, 0.12, and 0.10, with target gap $d_{i,t} = 0.20$. The first transition masks the first chunk, the second masks the second, and the third masks both the third and fourth chunks to reach the target. The four views retain relevance masses $(1.00, 0.66, 0.40, 0.10)$. This arithmetic example shows both nesting and chunk-level overshoot: the realized gaps are 0.34, 0.26, and 0.30, not exactly 0.20. Relevance-mass control specifies a selection threshold rather than an exact amount of removed evidence.

Normally $d_{i,t} = \delta_t$, the controller's global gap. The implementation selects a document-specific gap for diffuse relevance distributions: if $\max_j w_{i,j} < 0.02$, it uses

$$d_{i,t} = \min\left\{ \frac{\max\{1, \lfloor M_i/(K+1) \rfloor\}}{M_i},\ 0.5 \right\}. \tag{3}$$

Selection proceeds by relevance mass in both branches. The global controller governs the default branch, while the document-specific rule sets the threshold for diffuse distributions using the chunk count and candidate count.

For each masked chunk of n tokens, the renderer repeats the tokenization of `[REDACTED]` and truncates the filler to n tokens. This preserves the constructed token-array length and original chunk offsets. The arrays are then decoded to text and inserted into generation prompts, which are tokenized for model input. This construction keeps the original chunk locations visible and marks unavailable passages consistently across successive source views.

2.2 From Source Views to Candidate Rankings

A generator G_ϕ, held fixed during a generation cycle, samples one answer per view:

$$y_i^{(k)} \sim G_\phi(\cdot \mid x_i^{(k)}, q_i), \qquad k = 1, \ldots, K. \tag{4}$$

We assign the proxy order $y_i^{(1)} \succ \cdots \succ y_i^{(K)}$ by source-view construction index, along which retained evidence is non-increasing. If masking exhausts the document before all views are constructed, the remaining views share the same context; their labels still follow construction index. Thus the target is deterministic even at saturation, while its interpretation as answer quality is a modeling hypothesis. All candidates use the same decoding configuration, keeping the sampling protocol consistent across source views. The candidate set therefore links a common question to responses generated under explicitly controlled evidence conditions.

Generation and ranking deliberately use different information. Degraded views create candidate differences, while the original document provides a common basis for comparing them. The ranker is not shown the masked views or their source indices, so it must infer the assigned order from the displayed answers and document. In contrast, the generator receives an explanation of the redaction marker and is instructed to use available content. The redaction marker makes the intervention explicit to the generator.

To hide the source order, sample a display permutation σ_i and show candidate $y_i^{(\sigma_i(j))}$ at position j. The target order over displayed candidate IDs is

$$\pi_i^\star = (\sigma_i^{-1}(1), \ldots, \sigma_i^{-1}(K)). \tag{5}$$

The policy sees the original document x_i, question q_i, and shuffled answers. It returns a best-to-worst order $\hat{\pi}_i$ in a designated `\boxed{}` field. Randomizing the display order removes a fixed positional answer key. The verifier reconstructs the target through the stored permutation, giving each candidate set an exact, automatically recoverable source-view label.

For example, displaying the candidates in source order $(3, 1, 4, 2)$ assigns display ID 2 to the least-redacted candidate and display ID 3 to the most-redacted one. The target is therefore $(2, 4, 1, 3)$, as in Figure 2. The display permutation and the desired output are inverse mappings, not interchangeable lists. Only the verifier uses this source-to-display mapping; it is not supplied as part of the policy prompt.

2.3 Ordinal Reward and Policy Optimization

Candidate-pair formulation. For a valid permutation, let $p_i(a)$ and $p_i^\star(a)$ be candidate a's positions in the predicted and target orders. The natural ordinal reward is

$$r_{\text{pair}}(\hat{\pi}_i, \pi_i^\star) = \frac{1}{\binom{K}{2}} \sum_{a<b} \operatorname{sgn}[(p_i(a) - p_i(b))(p_i^\star(a) - p_i^\star(b))] \,. \tag{6}$$

This is standard candidate-pair Kendall agreement, with range $[-1, 1]$. It is invariant to a common relabeling of candidates. It expresses agreement by comparing the predicted and target positions of each candidate pair. This defines ordinal agreement with the source-view target.

Ranking reward. The evaluated training procedure computes Kendall correlation directly between the predicted and target candidate-ID lists:

$$r_{\text{rank}}(\hat{\pi}_i, \pi_i^\star) = \frac{1}{\binom{K}{2}} \sum_{u<v} \operatorname{sgn}\big[(\hat{\pi}_{i,u} - \hat{\pi}_{i,v})(\pi^\star_{i,u} - \pi^\star_{i,v})\big] \,. \tag{7}$$

Here u, v index positions in the candidate-ID lists. We denote this training signal by r_{rank} throughout the experimental analysis. Each term compares numerical order within the two lists, whereas Equation 6 compares positions indexed by candidate identity. Appendix A illustrates the two representations. Valid outputs receive a score in $[-1, 1]$; malformed rankings receive -2.

The two sampling counts have distinct roles: K is the number of answers to be compared, whereas G is the number of alternative rankings sampled for that same comparison. Increasing K changes the ranking task; increasing G changes the group used to estimate the policy update. For each ranking prompt, the policy samples G completions with rewards $r_{i,1}, \ldots, r_{i,G}$. Group-relative policy optimization (Shao et al., 2024) uses standardized within-prompt advantages

$$A_{i,g} = \frac{r_{i,g} - \mu_i}{s_i + \epsilon}, \qquad \mu_i = G^{-1} \sum_{g=1}^{G} r_{i,g}, \tag{8}$$

where s_i is the within-group reward standard deviation. A clipped policy-gradient objective updates the policy, with a separate KL loss against the reference policy. Group normalization centers and scales the feedback for alternative completions of the same prompt.

If every sampled ranking for a prompt receives the same reward, all group-relative advantages are zero. Such a group contributes no relative-reward policy-gradient signal, although the reference KL term remains active. The relative-reward signal is consequently determined by variation among the sampled rankings for a given prompt. This connects task construction to optimization: the controller adjusts evidence gaps as the policy's ranking behavior changes over training.

2.4 Feedback-Driven Curriculum

Let b_t be the mean implemented reward among valid-format training completions. The controller assigns $b_t = -1$ if all completions are invalid. Its state follows

$$e_t = \lambda e_{t-1} + (1 - \lambda) b_t, \tag{9}$$

with the global gap updated as

$$\delta_{t+1} = \begin{cases} \max\{\delta_{\min}, (1-\eta)\delta_t\}, & e_t > \theta_h, \\ \min\{\delta_{\max}, (1+\eta)\delta_t\}, & e_t < \theta_\ell, \\ \delta_t, & \text{otherwise.} \end{cases} \tag{10}$$

The dead band $[\theta_\ell, \theta_h]$ leaves the gap unchanged. Reducing the gap is intended to make adjacent candidates harder to distinguish; increasing it is intended to separate their evidence conditions. The controller applies this rule to the scalar relevance-gap state shared by the default construction branch. The logged gap is the *post-update* state, used in subsequent construction.

EMA smoothing reduces sensitivity to an isolated batch, while the interval between the two thresholds avoids changing the gap whenever feedback moves slightly. Multiplicative updates scale the

adjustment with the current gap. These choices regulate task construction rather than directly updating model weights: e_t changes the next batch's gap, whereas GRPO uses the current batch to update policy weights through the ranking rewards and reference regularization.

Task construction and policy optimization operate at complementary levels. The controller changes the evidence supplied to future candidate-generation calls; GRPO changes how the policy ranks the current candidates. The generator remains fixed within each generation cycle. A periodic policy-to-generator refresh is an optional extension of this interface, shown in Figure 2; Algorithm 1 specifies the evaluated controller-driven training procedure.

Complete training step. Algorithm 1 summarizes the evaluated training procedure. Relevance scores are prepared before this loop; candidates are generated for each document batch. The generator and KL reference serve different purposes: the former constructs answers, while the latter regularizes policy updates by penalizing divergence from the reference distribution.

Algorithm 1 Long-Zero training with a ranking reward

Inputs: scored documents, policy π_θ, generator G_ϕ, reference policy;
candidate count K, ranking group size G, controller state (δ_t, e_{t-1}).
For each document batch at step t:
1. Construct K nested views per document using $d_{i,t}$ (Eqs. 2–3).
2. Generate one answer per view with fixed G_ϕ.
3. Shuffle answers; build the full-document ranking prompt and target $\pi_i^\star$.
4. Sample G ranking completions for each prompt with π_θ.
5. Score valid rankings by Eq. 7; assign -2 to malformed outputs.
6. Update θ with group-relative advantages, clipping, and reference KL loss.
7. Aggregate valid rewards into b_t; update e_t and δ_{t+1} (Eq. 10).

3 Experiments

We evaluate how information-derived post-training transfers to long-context QA and passage reranking, and examine the evolution of ranking rewards and curriculum state during training. The two downstream task families test complementary uses of document evidence: answering questions and ordering relevant passages. Training trajectories provide a further view of the learning process behind these downstream comparisons.

3.1 Experimental Setup

Models and supervision. The two models are Qwen2.5-7B-Instruct and Qwen3-4B-Instruct-2507. "Base" denotes the corresponding instruction-tuned checkpoint before the additional post-training studied here, not a pretrained-only model. We compare Base, the available SFT checkpoints, and Long-Zero. Long-Zero does not use reference answers in its ranking reward; SFT uses target responses. The comparison evaluates how these post-training approaches affect downstream QA and evidence-based passage ordering.

Training. The representative 7B trajectory uses batch size 128, $K = 4$ candidates, $G = 4$ ranking completions, candidate-generation temperature 0.3, and KL coefficient 0.04. Its controller uses $\delta_0 = 0.30$, $\delta_{\min} = 0.05$, $\delta_{\max} = 0.40$, $\eta = 0.10$, $\lambda = 0.70$, and thresholds $(0.20, 0.50)$. Appendix B summarizes the documented settings for both 7B and 4B, with model scales labeled consistently throughout the paper. Tables and curves report the procedure trained using the ranking reward in Equation 7.

Evaluation and checkpoint selection. We evaluate long-context QA on LongBench v2 (Bai et al., 2024c), reporting Overall scores together with the Easy/Hard and Short/Medium/Long breakdowns. This organization allows us to examine aggregate performance alongside question difficulty and context length. Non-CoT sampling and CoT are separate evaluation conditions. HELMET (Yen et al., 2025) results cover MS MARCO passage reranking at 8k, 16k, 32k, and 64k tokens. Scores are NDCG@10 on a 0–100 scale, with an unweighted macro-average over lengths. The recorded protocol specifies 100 examples, two demonstrations, and greedy decoding. Reporting each context length alongside the macro score makes the contributions of shorter and longer inputs explicit.

Table 1: **Long-context QA results.** LongBench v2 results for 7B and 4B models. Non-CoT uses sampling, and CoT is compared separately. Higher is better. Bold marks the highest score within each model–decoding block, including ties. Overall and subset scores compare Base, SFT, and Long-Zero under each evaluation condition.

Method	Overall	Easy	Hard	Short	Medium	Long
7B / non-CoT						
Base	30.4	30.2	30.5	40.6	26.5	21.3
SFT	**32.6**	**36.0**	30.5	**42.2**	27.2	26.7
Long-Zero	32.4	34.9	**30.9**	40.0	**28.8**	**26.9**
7B / CoT						
Base	32.8	31.2	33.8	35.0	33.0	28.7
SFT	33.4	31.8	**34.4**	35.6	**33.5**	**29.6**
Long-Zero	**33.6**	**34.4**	33.1	**39.4**	32.1	26.9
4B / non-CoT						
Base	34.2	39.1	31.2	35.0	34.4	32.4
SFT	**35.8**	**40.6**	32.8	**37.8**	34.4	**35.2**
Long-Zero	35.4	39.1	**33.1**	36.7	**35.8**	32.4
4B / CoT						
Base	38.8	42.7	36.3	42.8	36.3	37.0
SFT	**41.9**	**48.4**	37.9	46.1	**40.9**	37.0
Long-Zero	41.6	44.3	**39.9**	**46.7**	38.1	**39.8**

Table 2: **HELMET MS MARCO reranking.** NDCG@10 on a 0–100 scale; Macro averages the four context lengths. Bold marks the best value at each model scale. Deltas are calculated from the displayed macro scores. Results cover both model scales under the same reported reranking protocol.

Model	Method	8k	16k	32k	64k	Macro	Δ Base
7B	Base	57.22	44.19	27.81	**15.76**	36.24	–
	SFT	58.12	44.43	27.13	11.41	35.27	-0.97
	Long-Zero	**63.14**	**48.31**	**35.92**	15.51	**40.72**	+4.48
4B	Base	74.24	59.20	41.28	13.11	46.96	–
	SFT	74.76	58.13	40.87	12.78	46.63	-0.33
	Long-Zero	**75.30**	**60.61**	**42.35**	**16.47**	**48.68**	+1.72

We report results by model scale, method, and decoding condition. All four QA conditions compare Base, SFT, and Long-Zero, with column maxima highlighted in Table 1. We examine QA transfer and passage reranking across context lengths.

3.2 Downstream Results

Long-context QA. Table 1 shows improvements over Base in all four model–decoding conditions: $+2.0$ and $+0.8$ points for 7B non-CoT and CoT, and $+1.2$ and $+2.8$ for 4B. Long-Zero also exceeds SFT on 7B CoT Overall (33.6 versus 33.4). Its gains extend to challenging subsets: on 4B CoT, Hard increases to 39.9 versus SFT's 37.9, and Long reaches 39.8 versus 37.0. With non-CoT decoding, Long-Zero achieves the highest Hard and Medium scores at both model scales. These results highlight complementary benefits across question difficulty, context length, and decoding conditions.

Transfer to passage reranking. Table 2 provides the clearest distinction between the evaluated post-training methods. Long-Zero increases macro NDCG@10 from 36.24 to 40.72 for 7B and from 46.96 to 48.68 for 4B. Relative to SFT, the gains are 5.45 and 2.05 points. Long-Zero achieves the highest macro score at each model scale. This pattern complements the QA results by showing transfer to the relative assessment of passages, where the model must identify and order evidence relevant to a query.

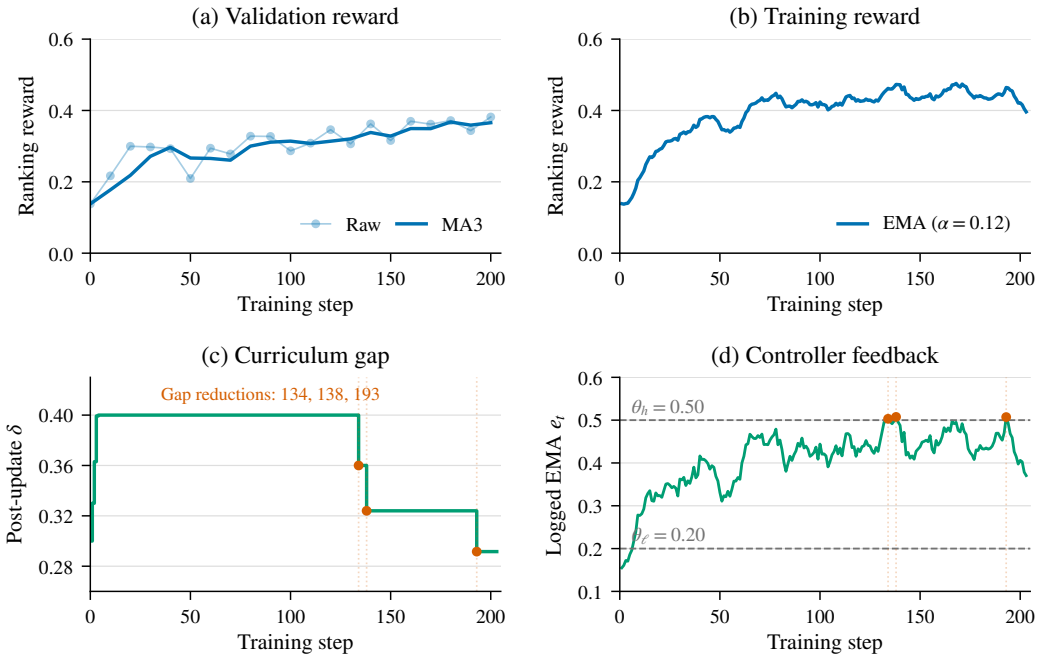

Figure 3: **7B training dynamics from one run.** (a) Raw validation ranking rewards with a trailing mean of up to three observations (MA3). (b) Training ranking reward, displayed with EMA smoothing ($\alpha = 0.12$). (c) Logged post-update global gap, including the configured initial state at step 0. (d) Logged controller EMA ($\lambda = 0.70$); dashed lines denote its thresholds. Orange markers identify the three gap reductions at steps 134, 138, and 193. Rewards are r_{rank} from Equation 7. The controller excludes invalid-format rewards; training and validation rewards include them. The four panels align the observed reward and controller trajectories by training step.

The context-length breakdown further characterizes these gains. The 4B Long-Zero model improves over Base at every evaluated length, including $+3.36$ points at 64k. The 7B improvements span 8k–32k, with the largest increase at 32k ($+8.11$). Relative to SFT, Long-Zero scores higher at every reported length for both scales. Candidate comparison during training and passage reranking during evaluation both require relative assessment under document evidence. The gains across these evaluation settings make this shared structure a useful perspective for interpreting the transfer results.

3.3 Training Dynamics

Figure 3 relates reward learning to the controller's state in a single 7B run. The raw validation reward rises from 0.1380 at step 0 to 0.3818 at step 200, with substantial intermediate variation. The first and last 20 training steps have raw mean rewards of 0.2689 and 0.4225, respectively. Validation uses the initial degradation setting but regenerates candidates; its variation can therefore reflect both candidate generation and changes in the policy performing the ranking task.

The gap first increases from 0.30 to 0.40 during steps 1–4 as the controller relaxes the curriculum. Later, feedback crosses the upper threshold of 0.50 at steps 134, 138, and 193, with EMA values 0.5031, 0.5075, and 0.5071. The corresponding post-update gaps are 0.36, 0.324, and 0.2916. These events show the controller moving from larger evidence separations toward progressively smaller gaps as feedback improves. The recorded threshold crossings and post-update values follow the update rule in Equation 10, connecting the observed curriculum trajectory directly to its feedback mechanism.

Checkpoint sensitivity. We examine four observed late checkpoints from one 4B training series. Their HELMET macro scores range from 48.08 to 48.68, all above Base (46.96) and the evaluated

SFT checkpoint (46.63). These observations show that the gain is present at multiple recorded checkpoints. The observed range characterizes reranking performance during later training.

4 Related Work

Long-context alignment and evaluation. LongAlign studies supervised long-context alignment using instruction–response demonstrations (Bai et al., 2024a). LongReward and LOGO investigate AI feedback and preference-based alignment (Zhang et al., 2024; Tang et al., 2024). Long-Zero obtains its ranking target from a controlled change in evidence availability. It combines corpus questions and pretrained relevance scores to construct source-view prediction tasks automatically. RULER, LongBench, LongBench v2, and HELMET assess complementary aspects of context use, from retrieval to realistic task performance (Hsieh et al., 2024; Bai et al., 2024b;c; Yen et al., 2025). We evaluate transfer through QA and HELMET passage reranking, measuring both answer generation and relative evidence assessment.

Verifiable rewards and ordinal supervision. DeepSeekMath and DeepSeek-R1 demonstrate reinforcement learning with automatically checked task outcomes (Shao et al., 2024; Guo et al., 2025). Long-Zero extends the use of automatically checkable targets to comparisons among free-form candidate responses. Its target is the source-view order assigned during construction, and verification scores the predicted ranking against that order. The supervision is therefore ordinal and operates on candidate sets, with the complete document providing the context for comparison. Section 2.3 specifies the candidate-pair formulation and the ranking reward, linking source-view labels to policy-optimization feedback.

Adaptive curricula and self-evolving training. Curriculum learning matches training experience to learner competence (Bengio et al., 2009; Graves et al., 2017). Absolute Zero and R-Zero explore reasoning systems with self-generated tasks (Zhao et al., 2025; Huang et al., 2026). Long-Zero retains corpus questions and adapts the evidence supplied during candidate generation. Its relevance gap gives the controller a continuous variable for adjusting this intervention through training feedback. The generator supplies candidate answers while the policy learns to compare them; periodic refresh is an optional extension. The evaluated adaptive mechanism is the controller, which translates ranking feedback into subsequent source-view construction.

5 Limitations

Source-view order provides a proxy for answer quality; redundant evidence and generation variability can produce ties or reversals. Chunk granularity and relevance scoring also determine the realized intervention.

Evaluation covers two model scales and two task families using selected-checkpoint point estimates. Repeated seeds, held-out checkpoint selection, matched-compute comparisons, and component ablations would clarify robustness and individual contributions. The evaluated mechanism combines the ranking reward in Equation 7 with curriculum control; candidate-pair rewards and successful generator refresh remain outside the demonstrated results.

Reproduction also requires verification of evaluator revisions, data splits and overlap, per-example scoring records, and checkpoint-specific training configurations.

6 Conclusion

Long-Zero turns controlled evidence degradation into automatically checkable ordinal supervision for long-context post-training. It combines nested source views, full-document candidate comparison, and feedback-driven adjustment of relevance gaps. The evaluated models improve over Base in all four QA conditions and at both scales on HELMET reranking, with additional gains over SFT on 7B CoT Overall and reranking macro scores. These results show that the information supplied during answer generation can serve as a useful resource for training models to compare and use long-context evidence.

AI USE STATEMENT

We used AI tools to assist with manuscript writing and language polishing, as well as coding tasks during the experimental workflow. The authors take full responsibility for the content, experimental results, and conclusions presented in this paper.

REFERENCES

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1376–1395, 2024a. doi: 10.18653/v1/2024.findings-emnlp.74.

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. 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)*, pp. 3119–3137, 2024b. doi: 10.18653/v1/2024.acl-long.172.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024c. URL https://arxiv.org/abs/2412.15204.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1311–1320, 2017. URL https://proceedings.mlr.press/v70/graves17a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=96apU6YzSO.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Zecheng Tang, Zechen Sun, Juntao Li, Qiaoming Zhu, and Min Zhang. LOGO – long cOntext aliGnment via efficient preference Optimization. *arXiv preprint arXiv:2410.18533*, 2024. URL https://arxiv.org/abs/2410.18533.

Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat, and Danqi Chen. HELMET: How to evaluate long-context models effectively and thoroughly. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=293V3bJbmE.

Jiajie Zhang, Zhongni Hou, Xin Lv, Shulin Cao, Zhenyu Hou, Yilin Niu, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. LongReward: Improving long-context large language models with AI feedback. *arXiv preprint arXiv:2410.21252*, 2024. URL https://arxiv.org/abs/2410.21252.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute Zero: Reinforced self-play reasoning with zero data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=neZSGqhxDa.

A Implementation Semantics

Sequence and position representations. For $K = 4$, consider a predicted order $(1, 3, 4, 2)$ and target order $(2, 3, 4, 1)$. Equation 7 computes Kendall correlation between these lists and returns $+2/3$. Converting each list to positions indexed by candidate identity gives candidate-pair agreement $-2/3$: only one of six candidate pairs is concordant. Across all $4! \times 4! = 576$ pairs of orders, the two formulas disagree for 296 pairs. This exhaustive calculation characterizes the relationship between the two representations over the complete four-candidate permutation space.

The position representation associates each candidate identity with its location in the ranking. The sequence representation associates each list position with the candidate ID placed there. Inverting a permutation converts between the two. Equation 6 operates on the position representation, while Equation 7 operates on the candidate-ID sequence. This distinction specifies exactly which objects are compared in each reward calculation.

Formatting and controller aggregation. A valid ranking must contain every candidate ID exactly once. Invalid outputs receive -2. Training and validation mean rewards include this penalty. The controller averages only rewards at least -1, using -1 when no valid completion exists. Its EMA therefore differs from a smoothed all-completion reward even when the same smoothing coefficient is used. The training display EMA in Figure 3 uses new-observation weight 0.12; the controller instead assigns weight $1 - \lambda = 0.30$ to each new batch-level observation of valid-format reward.

Mask rendering and construction state. The masker's diffuse-distribution branch uses Equation 3. In either branch, a shared pointer advances through chunks sorted by relevance, and the selected chunks are added to the cumulative masked set. The renderer repeats the tokenization of the redaction marker to fill each selected chunk's original token positions. The resulting token arrays are decoded, inserted into generation prompts, and tokenized for inference.

The construction state consists of the current global gap and the controller EMA. Each batch produces ranking completions and rewards, then updates this state for the next batch. Generator weights are fixed during a generation cycle; the optional refresh interface operates between cycles. The KL reference is a separate model role, supplying the distribution used to regularize the policy during optimization.

B Training Configuration

Table 3 reports the controller and sampling settings documented for the 7B trajectory and the 4B training configuration. The shared batch size, candidate count, ranking group size, generation temperature, and KL coefficient specify the common training structure. The scale-specific gap bounds and controller coefficients define how feedback is translated into changes in evidence separation. Section 5 describes the scope of these configuration records.

Table 3: Documented controller and sampling settings by model scale. The 7B column corresponds to the plotted trajectory; the 4B column summarizes its training configuration.

Parameter	7B	4B
Document batch size	128	128
Candidates K / ranking group G	4 / 4	4 / 4
Generator temperature	0.3	0.3
KL coefficient	0.04	0.04
Initial gap δ_0	0.30	0.30
Gap bounds	[0.05, 0.40]	[0.05, 0.45]
Gap step ratio η	0.10	0.08
Controller decay λ	0.70	0.85
Thresholds (θ_ℓ, θ_h)	(0.20, 0.50)	(0.20, 0.50)