COPY LESS, GROUND MORE: OVERCOMING REPETITIVE COPYING IN LONG-CONTEXT REASONING VIA EVIDENCE-AWARE REINFORCEMENT LEARNING

Lizhe Fang[1] **Weizhou Shen**[2] **Tianyi Tang**[2] **Yisen Wang**[1*]
[1] Peking University
[2] Alibaba Group

ABSTRACT

Large language models that generate step-by-step reasoning traces have achieved strong performance on complex tasks, and extending them to long-context settings has emerged as an important frontier. However, we identify a critical failure mode in this regime: *repetitive copying*, where models extensively copy text from the input into their reasoning traces rather than productively solving the problem. We show that this behavior is pervasive across frontier long-context LLMs and intensifies with context length. By separating each prompt into task-relevant key evidence and irrelevant distractor context, we further show that the root cause is insufficient grounding: models copy from the prompt indiscriminately, and those that fail to focus on key evidence are far more likely to answer incorrectly. Motivated by this diagnosis, we propose GEAR (Grounding Evidence-Aware Reward), a reward shaping method that augments the accuracy signal with a grounding reward for overlap with key evidence and a distractor penalty for overlap with irrelevant context. To enable GEAR on natural-language data, we develop an automated pipeline that constructs evidence-annotated training data from arbitrary documents. We validate GEAR across multiple model scales and benchmarks, showing consistent improvements of up to +4.6 average points over standard RL with accuracy-based rewards, with larger gains at longer contexts, while also reducing repetitive copying and thinking length. Our findings suggest that, even as long-context evaluation shifts from simple retrieval toward complex reasoning, accurate grounding in relevant evidence remains an indispensable capability with substantial room for improvement.

1 INTRODUCTION

Recent large language models (LLMs) equipped with explicit thinking capabilities have demonstrated remarkable performance on complex reasoning tasks. By generating step-by-step reasoning traces before arriving at a final answer, thinking models such as the Qwen3 series (Yang et al., 2025), the Claude series, and the DeepSeek series (Guo et al., 2025) consistently outperform their non-thinking counterparts. Meanwhile, the evaluation of long-context LLMs has increasingly shifted from simple retrieval tasks toward complex reasoning over extended inputs (Bai et al., 2025; Hsieh et al., 2024; Chen et al., 2026), making these thinking models a natural fit for the frontier of long-context reasoning.

Despite this progress, we identify a critical failure pattern when thinking models are applied to *long-context reasoning*: **repetitive copying**. When reasoning over long inputs, models frequently copy large passages directly from the prompt into their thinking traces. This behavior is pervasive, appearing across seven frontier long-context LLMs that include both proprietary APIs and open-weight checkpoints, and it becomes more severe as context length increases. Crucially, repetitive copying not only reduces accuracy but also substantially increases reasoning length, as the model spends its token budget reproducing prompt content rather than solving the task.

*Corresponding Author: Yisen Wang (yisen.wang@pku.edu.cn).

To understand the root cause of this behavior, we conduct a systematic analysis and find that the core issue is **insufficient grounding**: models copy prompt content indiscriminately because they fail to distinguish task-relevant evidence from irrelevant distractor context. We quantify how selectively a model copies from task-relevant vs. irrelevant content and show that correctly answered samples exhibit significantly more selective copying than incorrect ones. These findings suggest that copying from the prompt is not inherently harmful. The problem instead lies in indiscriminate copying: models that focus on key evidence are more likely to succeed, whereas those that copy broadly from the context are more likely to fail.

Motivated by this diagnosis, we propose **GEAR** (**G**rounding **E**vidence-**A**ware **R**eward), a reward shaping method that augments the standard accuracy signal with two complementary components. A *grounding reward* encourages the model to engage with key evidence by measuring n-gram overlap between the reasoning trace and annotated support spans. A *distractor penalty* discourages indiscriminate copying by penalizing overlap with irrelevant context. To extend GEAR beyond synthetic benchmarks where evidence annotations are readily available, we further develop an automated three-stage pipeline that constructs evidence-annotated QA pairs from arbitrary long documents.

We train three Qwen3.5 models across different scales with GSPO (Zheng et al., 2025) using the GEAR reward on a mixture of 3.2k long-context samples from both synthetic tasks and natural-language documents. On five held-out benchmarks, GEAR consistently improves over standard RL with accuracy-based rewards by up to +4.6 average points, while simultaneously reducing repetitive copying and thinking length.

Our contributions are as follows:

- We identify the *repetitive copying* problem in long-context LLM reasoning, showing that it is prevalent across models and negatively correlated with accuracy. We further trace its root cause to insufficient evidence grounding, as models copy indiscriminately instead of selectively engaging with task-relevant information.
- We propose GEAR, a simple and effective reward shaping method that combines a grounding reward with a distractor penalty to encourage selective evidence engagement during RL training.
- We develop an automated data construction pipeline that generates evidence-annotated long-context QA pairs from arbitrary document corpora, enabling GEAR to scale beyond synthetic benchmarks to natural-language data.
- We validate GEAR on five diverse benchmarks, showing consistent improvements over standard RL with accuracy-based rewards and confirming that the gains stem from reduced indiscriminate copying and improved grounding.

2 Related Work

Reinforcement learning for reasoning. The success of DeepSeek-R1 (Guo et al., 2025) demonstrated that strong reasoning capabilities can emerge from RL training with verifiable rewards alone, without supervised fine-tuning. This reinforcement learning with verifiable rewards (RLVR) paradigm has since been refined along several axes. GRPO (Shao et al., 2024) replaces the critic network with group-relative advantage estimation, reducing memory overhead. DAPO (Yu et al., 2025a) introduces asymmetric clipping and dynamic sampling for more stable long-CoT training. GSPO (Zheng et al., 2025) shifts from token-level to sequence-level importance ratios, addressing variance issues in MoE models. On the data side, s1 (Muennighoff et al., 2025) shows that as few as 1,000 curated traces can activate latent reasoning ability, while RLPR (Yu et al., 2025b) uses the model's own generation probability as a graded reward to extend RLVR beyond math domains. These works focus primarily on short-context reasoning; our work addresses the distinct challenges that arise when RL is applied to long-context tasks.

Long-context RL training. Several concurrent works extend RLVR to long-context settings. LongR-LVR (Chen et al., 2026) proves that outcome-only rewards cause vanishing gradients for contextual grounding and augments the answer reward with a verifiable context reward based on chunk-level F-

scores. GoLongRL (Lv et al., 2026) takes a capability-oriented approach, constructing a 23K-sample dataset spanning 9 task types with heterogeneous reward functions and proposing task-level reward normalization. QwenLong (Wan et al., 2025; Shen et al., 2025) combines curriculum-guided RL with difficulty-aware sampling, while LongR (Ping et al., 2026) introduces a contextual density reward that measures information gain from cited context. LoongRL (Wang et al., 2025) pads short-context multi-hop QA with distractors to create long-context training data. Our work differs from these approaches in both diagnosis and method: we identify *repetitive copying* as a specific failure mode of long-context reasoning and design a reward that directly targets this behavior by distinguishing key evidence from distractor context at the n-gram level, rather than operating on coarse chunk identifiers.

Repetition and overthinking in LLMs. Repetitive degeneration in neural text generation has been studied from multiple angles. Li et al. (2023) trace degeneration to repetitive training data, while Yao et al. (2025) identify specific "repetition features" in model activations. Mahaut & Franzon (2025) show that repetition arises from distinct mechanisms including uncertainty and overconfidence. In the context of reasoning models, the "overthinking" problem, in which models generate excessively long reasoning traces, has received growing attention (Chen et al., 2024; Dai et al., 2026), with Wu et al. (2025) demonstrating an inverted-U relationship between reasoning length and accuracy. Our work connects these threads by showing that in long-context settings, the dominant form of excessive reasoning is not aimless elaboration but *direct copying from the prompt*, and that this behavior is directly linked to insufficient evidence grounding. Separately, Fang et al. (2025) show that not all tokens contribute equally to long-context understanding, suggesting that the impact of repetitive generation may depend on *what* content is being repeated rather than repetition alone.

3 The Repetitive Copying Issue in Long-Context Reasoning

In this section, we present a systematic empirical study of repetitive copying in thinking LLMs. Our analysis proceeds in three stages. First, we quantify how prevalent this behavior is across models and context lengths (Section 3.2). Second, we establish that higher copying rates correlate with lower task accuracy and inflated reasoning length (Section 3.3). Third, we investigate the root cause by examining *what* the model copies (Section 3.4).

3.1 Experimental Setup

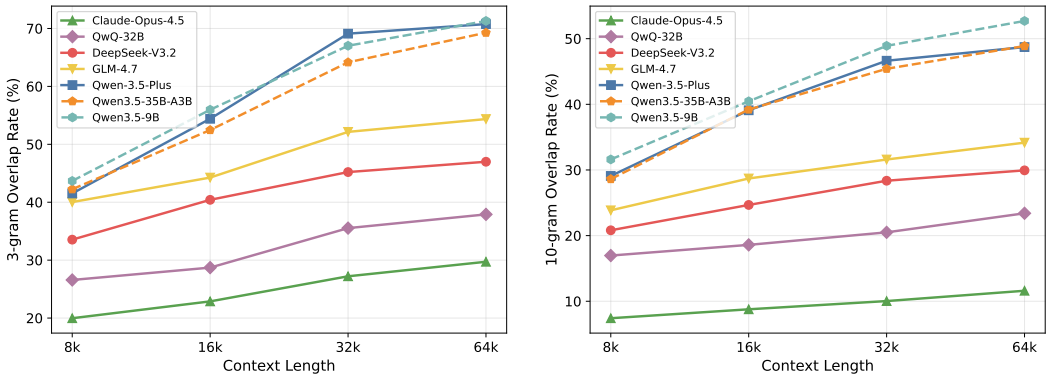
Metrics. We quantify repetitive copying using *overlap rate*, which measures how much of a model's output is copied directly from the input prompt. We define $y = (y_1, \ldots, y_m)$ as the reasoning trace (model output) and collect all distinct n-grams from the prompt x into a lookup set $\mathcal{N}_n(x)$. We then scan through y with a sliding window of size n: for each position i, the window $(y_i, \ldots, y_{i+n-1})$ is marked as a match if it appears in $\mathcal{N}_n(x)$. The overlap rate is the match rate across all positions:

$$\text{Overlap}_n(y\|x) = \frac{1}{m-n+1} \sum_{i=1}^{m-n+1} \mathbf{1}[(y_i, \ldots, y_{i+n-1}) \in \mathcal{N}_n(x)]\,. \tag{1}$$

We report results for both $n = 3$ and $n = 10$: 3-grams capture overall copying volume, while 10-grams isolate longer contiguous spans that unambiguously indicate deliberate transcription.

Models. We evaluate seven thinking LLMs spanning proprietary APIs and open-weight checkpoints: Claude-Opus-4.5, DeepSeek-V3.2 (Liu et al., 2025), Qwen-3.5-Plus (Yang et al., 2025), GLM-4.7 (Glm et al., 2024), QwQ-32B (Team, 2025), Qwen3.5-35B-A3B, and Qwen3.5-9B. All models produce explicit reasoning traces before generating a final answer, allowing direct inspection of their intermediate reasoning for copying behavior.

Task. We study repetitive copying on GSM-Infinite (Zhou et al., 2025), a procedurally generated long-context arithmetic reasoning benchmark that extends GSM8K by embedding word problems within a large body of numerical descriptions. Each problem requires 10–20 sequential operations, and the model must identify the relevant numerical conditions scattered across the context and chain them correctly. Because problems and filler content are generated programmatically, both the character positions of the relevant conditions (support indices) and the ground-truth reasoning chains



(a) $n = 3$ (b) $n = 10$

Figure 1: n-gram overlap rate on GSM-Infinite as a function of context length across seven models. All models copy more as context grows, and the high 10-gram overlap confirms that copied spans are long contiguous passages.

are known exactly, enabling precise measurement of *what* the model copies, i.e., whether it targets task-relevant evidence or merely reproduces filler text. We construct datasets at context scales of 8k, 16k, 32k, and 64k tokens.

3.2 Repetitive Copying is Prevalent among LLMs

We begin by measuring how much reasoning models copy from the input and how this behavior changes with context length.

Figure 1 reports overlap rates on GSM-Infinite across four context scales (8k–64k). All models exhibit substantial copying even at the shortest context. At 8k, 3-gram overlap rates range from 20.0% (Claude) to 43.7% (Qwen3.5-9B), indicating that a significant portion of reasoning traces consists of copied prompt content. The problem worsens uniformly as context grows: Qwen-3.5-Plus increases from 41.5% to 70.8% at 64k, and even the least affected model (Claude) rises from 20.0% to 29.7%. The trend is consistent across all seven models, suggesting that longer inputs systematically elicit more copying during reasoning.

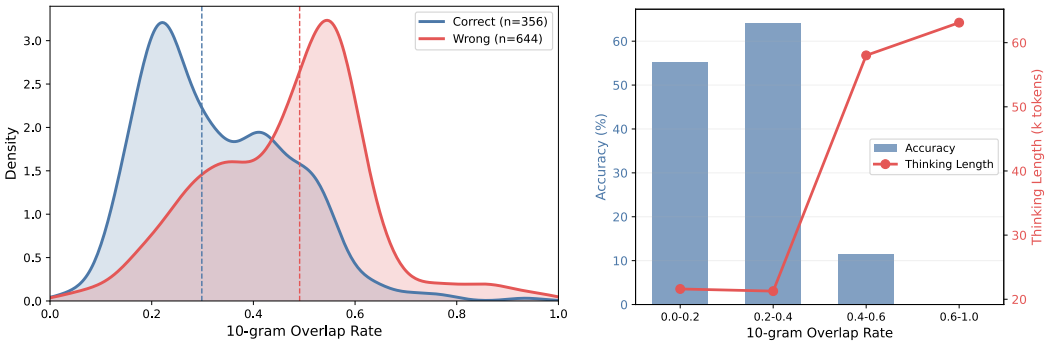
Comparing the 3-gram and 10-gram panels reveals that a significant fraction of copied content consists of long contiguous spans. The 10-gram overlap of Qwen3.5-9B reaches 52.7% at 64k, meaning that over half of all 10-gram windows in the reasoning trace are direct copies from the prompt, providing strong evidence of deliberate transcription rather than incidental lexical overlap.

3.3 Repetitive Copying Harms Task Performance

Having established that repetitive copying is widespread across all models (Section 3.2), we now examine its relationship with task performance. We use DeepSeek-V3.2 on GSM-Infinite at 64k context as a representative case study; results for additional models are provided in Appendix A.1.

Higher overlap correlates with lower accuracy and longer reasoning. We compare the 10-gram overlap distributions of correctly and incorrectly answered samples (Figure 2a). The two groups are clearly separated: correctly answered samples exhibit far less copying from the prompt than incorrect ones. This suggests that excessive copying is not a productive strategy: rather than helping the model locate relevant information, it floods the reasoning trace with unreprocessed prompt content.

The damage becomes even clearer when we examine the joint effect on accuracy and reasoning length (Figure 2b). Below an overlap rate of 0.4, accuracy remains at 55–64% and thinking length stays around 21k tokens. Beyond this threshold, accuracy collapses to 11% and eventually 0%, while thinking length nearly triples to over 58k tokens. In other words, the model spends its token budget transcribing the input rather than reasoning about it, and produces longer but less useful traces as a



(a) Overlap rate distributions

(b) Accuracy and thinking length vs. overlap

Figure 2: Repetitive copying harms both accuracy and reasoning efficiency (DeepSeek-V3.2 on GSM-Infinite, 64k context, 10-gram overlap). (a) Correctly answered samples exhibit markedly lower overlap rates. (b) As overlap rate increases, accuracy drops and thinking length (in tokens) grows sharply.

result. Importantly, this relationship is not merely a confound of problem difficulty: the same pattern holds when we group samples by the number of required reasoning steps (Appendix A.4).

3.4 Repetitive Copying Reflects Insufficient Grounding

Section 3.3 establishes that repetitive copying correlates with lower accuracy, but it does not explain *why*, since copying from the prompt is not inherently harmful. If the model selectively retrieves task-relevant information, it could even be beneficial. The critical question, then, is not *how much* the model copies, but *what* it copies: does it focus on key evidence, or does it reproduce prompt content indiscriminately?

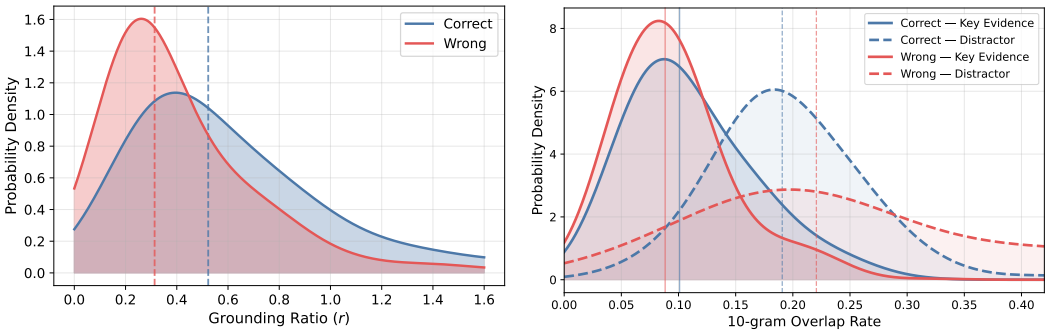
Measuring grounding with the key/distractor overlap ratio. To answer this question, we leverage the procedural construction of GSM-Infinite, where ground-truth support indices identify which portions of the prompt contain the key evidence needed to solve each problem. We split each prompt into key evidence (the annotated support spans) and distractor context (the remainder), and compute the overlap against each part separately. We use $n = 10$ for this analysis, as longer n-grams more reliably indicate deliberate transcription rather than incidental lexical overlap (Section 3.2). The *grounding ratio* is defined as:

$$r = \frac{\text{Overlap}_{10}(y \| x^{\text{key}})}{\text{Overlap}_{10}(y \| x^{\text{dist}})} \tag{2}$$

A higher grounding ratio indicates that the model's copying is more focused on task-relevant evidence.

Figure 3a shows the distribution of grounding ratios for correctly vs. incorrectly answered samples. Correctly answered samples exhibit markedly higher grounding ratios, indicating that successful reasoning involves more selective copying from task-relevant evidence. Figure 3b decomposes this further: correctly answered samples show both higher key evidence overlap and lower distractor overlap, confirming that accurate reasoning involves selectively engaging with relevant evidence while avoiding irrelevant context. We observe this pattern across five of the six additional models tested (Appendix A.2).

These results indicate that the model's long-context reasoning suffers not because it copies from the prompt *per se*, but because it fails to ground its copying in the relevant evidence. In summary, our analysis identifies a prevalent failure mode in long-context reasoning: models extensively copy irrelevant passages from the input into their reasoning traces, crowding out productive problem-solving.



(a) Grounding ratio

(b) Key evidence vs. distractor overlap

Figure 3: Copying behavior of correctly vs. incorrectly answered samples (DeepSeek-V3.2 on GSM-Infinite, 64k context). (a) Probability density of grounding ratio r (Eq. 2). Correctly answered samples exhibit higher grounding ratios, indicating more selective copying. (b) 10-gram overlap with key evidence (solid) and distractor context (dashed). Correctly answered samples show higher key evidence overlap and lower distractor overlap, confirming more selective copying from task-relevant evidence. Vertical lines indicate medians.

4 GEAR: Grounding Evidence-Aware Reward

Our analysis identifies insufficient grounding as the root cause of harmful repetitive copying (Section 3.4). We propose **GEAR** (**G**rounding **E**vidence-**A**ware **R**eward), a reward shaping method that augments the standard accuracy signal with a grounding reward and a distractor penalty to directly encourage selective evidence engagement during RL training. We first describe the reward design (Section 4.1), then present an automated data construction pipeline that produces the evidence annotations GEAR requires (Section 4.2), and finally detail the training setup (Section 4.3).

4.1 Reward Design

Since correctly answered samples exhibit systematically higher overlap with key evidence and lower overlap with distractor context (Section 3.4), we design two complementary reward signals that directly target this distinction. Given a prompt x with ground-truth support indices that partition it into key evidence x^{key} and distractor context x^{dist}, and a model-generated reasoning trace y, the GEAR reward is:

$$R(x, y) = R_{\text{acc}} + \alpha \cdot \text{Overlap}_n(y \| x^{\text{key}}) - \beta \cdot \text{Overlap}_n(y \| x^{\text{dist}}) \tag{3}$$

where $R_{\text{acc}} \in \{0, 1\}$ is the binary accuracy reward, $\text{Overlap}_n(\cdot)$ is the n-gram overlap rate defined in Eq. 1, and α, β are non-negative coefficients. In practice, n-grams that appear in both x^{key} and x^{dist} are excluded from the distractor term to avoid penalizing legitimate references to key evidence.

The grounding reward $R_{\text{ground}} = \alpha \cdot \text{Overlap}_n(y \| x^{\text{key}})$ encourages the model to selectively engage with task-relevant evidence, while the distractor penalty $R_{\text{dist}} = \beta \cdot \text{Overlap}_n(y \| x^{\text{dist}})$ discourages indiscriminate copying of irrelevant context. The two forces are complementary: the grounding term alone cannot prevent the model from copying *everything* (which would also yield high key overlap), and the distractor penalty alone cannot encourage the model to engage with key evidence at all. Notably, this reward requires no external verifier or retrieval module, as it operates entirely on n-gram statistics computable from the existing support annotations.

4.2 Evidence-Annotated Data Construction

While the support annotations required by GEAR are straightforward for synthetic benchmarks, extending to natural-language data requires an automated pipeline. We describe two complementary data sources below.

Structured synthetic data. We select two procedurally generated benchmarks that require distinct reasoning skills over long contexts. GSM-Infinite (Zhou et al., 2025) extends GSM8K into a long-context setting by embedding arithmetic word problems within large bodies of numerical descriptions; the model must locate relevant numerical conditions scattered across the context and chain 10–20 sequential operations. PhantomWiki (Gong et al., 2025) constructs fictional encyclopedias with interlinked entity articles, requiring multi-hop entity tracking to answer questions whose evidence spans multiple articles. We choose these two tasks to cover complementary reasoning types (numerical chaining and entity tracking) so that GEAR learns general evidence-grounding behavior rather than task-specific shortcuts. Because both benchmarks are generated programmatically, their generators natively produce support indices as part of the construction logic, requiring no additional annotation effort.

Natural-language data: a three-stage synthesis pipeline. To extend GEAR beyond synthetic benchmarks, we develop an automated pipeline that generates evidence-annotated QA pairs from arbitrary long documents. The key idea is to reverse the usual annotation workflow: rather than generating a question and then labeling which parts of the document support it, we first sample a small subset of document chunks as the designated evidence region, then constrain question generation to this region alone. This way, support annotations are obtained by construction. Given a source document d, the pipeline proceeds as follows:

1. **Document analysis and filtering.** A language model analyzes d to identify its informational characteristics (factual claims, numerical data, causal relationships, comparisons) and determines which question types are appropriate. Documents unsuitable for QA (e.g., structured data, product listings) are filtered out, and the identified question types guide the subsequent generation step.
2. **Evidence-constrained QA generation.** The document is segmented into text chunks (1,024 characters each). A random subset of 1–3 chunks is sampled as the *constraint context*, defining the key evidence region; the remainder constitutes the distractor context. A language model then generates a question answerable from the constraint context, along with 1–3 exact evidence quotations and a concise answer.
3. **Answer verification.** The generated question is independently re-answered using *only* the constraint context. We retain only pairs where this re-derived answer matches the reference answer, ensuring that the question is unambiguously answerable from the designated evidence.

We restrict answers to single-valued responses (a number or entity phrase, excluding yes/no questions) to support automated reward computation.

Training set composition. Using these two pipelines, we construct a training set of 3,200 problems: 1,000 each from GSM-Infinite and PhantomWiki (structured synthetic), plus 1,200 QA pairs generated from RedPajama-v2 (Weber et al., 2024) documents using the natural-language pipeline. The inclusion of both data types is intended to verify that GEAR generalizes across data sources rather than being limited to synthetic benchmarks.

4.3 Training Setup

Base model and algorithm. We train three base models: Qwen3.5-9B, Qwen3.5-27B, and Qwen3.5-35B-A3B (a Mixture-of-Experts variant) using Group Sequence Policy Optimization (Zheng et al., 2025) with the GEAR reward defined in Eq. 3. We set $n = 3$ for n-gram computation, rather than the $n = 10$ used in the diagnostic analysis (Section 3.4), as shorter n-grams provide denser reward signal that is more effective for RL optimization, and our ablation study confirms this is the optimal setting. We set $\alpha = 0.1$ and $\beta = 0.3$, penalizing distractor copying more aggressively than rewarding key evidence overlap; our sensitivity analysis validates this choice.

Training details. We train for 1 epoch ($\sim$100 optimizer steps) with a learning rate of 2×10^{-6}, a batch size of 32, and a rollout group size of 8. The maximum prompt length is 32k tokens and the maximum generation length is 16k tokens. Training data context lengths are uniformly distributed between 16k and 32k tokens.

5 EXPERIMENTS

5.1 MAIN RESULTS

Setup. We evaluate on five long-context benchmarks that are disjoint from the GEAR training data: Ruler (Hsieh et al., 2024), a synthetic retrieval benchmark testing multi-key and multi-value extraction across varying context lengths; LongBench-v2 (Bai et al., 2025), a comprehensive benchmark covering diverse long-context understanding tasks; BrowseComp-LC (Wei et al., 2025), a web browsing comprehension benchmark requiring synthesis of information across long documents; GraphWalks (OpenAI, 2025), a graph-structured reasoning benchmark that requires models to follow multi-hop paths and preserve intermediate state over long textual graph descriptions; and AA-LCR (Artificial Analysis, 2025), a long-context reasoning benchmark. All benchmarks natively use 128k-token contexts. We report results at two scales: a *32k* subset containing only tasks whose context fits within 32k tokens, and the full *128k* set. Because all AA-LCR instances exceed 32k tokens, this benchmark appears only in the 128k column. For each base model, we compare four configurations: (1) the pretrained Qwen3.5 base model, (2) GSPO trained with accuracy-only reward, (3) GSPO with the grounding reward added ($+R_{\text{ground}}$), and (4) GSPO with the full GEAR reward ($+R_{\text{ground}} + R_{\text{dist}}$).

GEAR yields consistent gains across benchmarks and model scales. Table 1 presents results across three model scales and two context lengths. GSPO + GEAR achieves the highest average score in all six model–context combinations. At 32k context, GEAR improves over accuracy-only GSPO by +2.8 (9B), +2.1 (35B-A3B), and +1.5 (27B) points on average. At 128k, the gains are larger: +4.6, +2.8, and +2.1, respectively. This suggests that evidence-aware reward shaping becomes more valuable as context length increases and locating relevant evidence becomes harder. Notably, GEAR is trained on contexts of 16k–32k tokens, yet the improvements at 128k ($4\times$ longer than the training distribution) are consistently larger than at 32k, demonstrating that the evidence-grounding behavior learned by GEAR generalizes robustly to unseen context lengths.

Both reward components are indispensable. A surprising finding emerges from the ablation rows. Adding the grounding reward *without* the distractor penalty ($+R_{\text{ground}}$) consistently *hurts* performance relative to accuracy-only GSPO, with drops of up to 5.1 points at 32k and 8.3 points at 128k. The degradation is especially severe on GraphWalks and BrowseComp-LC. Conversely, Table 3 we demonstrate that training with only the distractor penalty ($\alpha = 0$, $\beta = 0.3$) also yields an results worse than the baseline. Without a grounding signal to direct the model toward key evidence, the penalty merely discourages all copying, suppressing both useful and harmful engagement with the prompt. Only when both components are combined does the model learn to discriminate between relevant and irrelevant context, yielding the consistent gains of full GEAR.

5.2 EFFECT ON REPETITIVE COPYING AND GROUNDING

The main results demonstrate that GEAR improves task accuracy, but do these gains stem from the intended mechanism, i.e., reducing indiscriminate copying and improving evidence grounding? Table 2 reports answer-input overlap (token-level 3-gram) and thinking length (in tokens) for the base model, GSPO, GSPO + R_{ground}, and GSPO + GEAR on Ruler and LongBench-v2.

The grounding reward alone increases copying. Adding R_{ground} without the distractor penalty increases overlap relative to GSPO: 36.6% vs. 35.7% on Ruler and 28.5% vs. 27.2% on LongBench-v2. Thinking length also increases substantially (e.g., 5727 vs. 2808 tokens on Ruler). This reveals a failure mode: without a penalty for distractor copying, the grounding reward incentivizes the model to copy more from all parts of the input, including irrelevant context, resulting in longer and more repetitive reasoning that ultimately hurts task accuracy (Table 1).

GEAR suppresses copying through the distractor penalty. Only when the distractor penalty R_{dist} is added does the overlap drop below all other configurations: 27.0% on Ruler and 22.6% on LongBench-v2. Even on LongBench-v2, where GSPO alone actually increases overlap from 24.9% (base) to 27.2%, GEAR reduces it to 22.6%. The trajectory Base $\rightarrow$ GSPO $\rightarrow +R_{\text{ground}} \rightarrow$ +GEAR (overlap: 24.9 $\rightarrow$ 27.2 $\rightarrow$ 28.5 $\rightarrow$ 22.6 on LongBench-v2) illustrates that standard RL and

Table 1: Performance on long-context benchmarks under 32k and 128k contexts. All benchmarks are disjoint from the training data. Best results per context and model are in **bold**.

Model	Method	Ruler	LB-v2	Bcomp-LC	Graphwalks	AA-LCR	Avg
32k Context							
Qwen3.5-9B	-	77.4	50.5	61.7	85.0	-	68.7
	GSPO	91.5	55.9	77.7	88.9		78.5
	GSPO + R_{ground}	87.4	**58.6**	73.9	73.8		73.4
	GSPO + GEAR	**96.4**	**58.6**	**79.1**	**90.9**		**81.3**
Qwen3.5-35B-A3B	-	81.4	61.3	66.6	85.6	-	73.7
	GSPO	93.1	60.4	**75.6**	91.5		80.2
	GSPO + R_{ground}	90.7	**62.1**	73.0	88.3		78.5
	GSPO + GEAR	**96.2**	61.3	**75.6**	**95.9**		**82.3**
Qwen3.5-27B	-	82.9	60.4	72.5	85.0	-	75.2
	GSPO	95.5	60.4	74.9	91.5		80.6
	GSPO + R_{ground}	92.9	58.2	73.2	88.6		78.2
	GSPO + GEAR	**96.5**	**62.2**	**77.7**	**91.8**		**82.1**
128k Context							
Qwen3.5-9B	-	54.9	49.8	54.0	80.5	57.0	59.2
	GSPO	84.1	52.2	69.5	86.0	59.0	70.2
	GSPO + R_{ground}	79.6	52.6	57.6	60.8	59.0	61.9
	GSPO + GEAR	**90.8**	**54.7**	**71.7**	**88.8**	**68.0**	**74.8**
Qwen3.5-35B-A3B	-	74.6	54.0	57.8	82.8	59.0	65.6
	GSPO	86.4	55.4	**68.8**	89.6	67.0	73.4
	GSPO + R_{ground}	79.0	**58.4**	66.2	85.5	67.0	71.2
	GSPO + GEAR	**93.9**	57.6	68.3	**91.3**	**70.0**	**76.2**
Qwen3.5-27B	-	78.0	58.6	66.6	84.2	70.0	71.5
	GSPO	92.1	58.2	70.5	90.7	65.0	75.3
	GSPO + R_{ground}	89.3	57.1	67.8	87.2	63.0	72.9
	GSPO + GEAR	**93.3**	**60.0**	**70.9**	**91.0**	**72.0**	**77.4**

Table 2: Effect of GEAR on repetitive copying and thinking length (Qwen3.5-35B-A3B, 32k context). Overlap rate is computed as token-level 3-gram overlap rate (Eq. 1). $+R_{\text{ground}}$ adds the grounding reward only; +GEAR adds both grounding reward and distractor penalty. ↓: lower is better.

Benchmark	Metric	Base	GSPO	+ R_{ground}	+ GEAR
Ruler	Overlap Rate (↓)	36.9%	35.7%	36.6%	**27.0%**
	Think Length (tokens)	4290	2808	5727	2066
LongBench-v2	Overlap Rate (↓)	24.9%	27.2%	28.5%	**22.6%**
	Think Length (tokens)	5657	4677	6833	3989

grounding-only reward both fail to address repetitive copying; only the combined GEAR reward, with its explicit distractor suppression, consistently reduces it.

Thinking becomes more concise. Compared to GSPO, GEAR reduces thinking length from 2808 to 2066 tokens on Ruler and from 4677 to 3989 on LongBench-v2. In contrast, $+R_{\text{ground}}$ dramatically *increases* thinking length, more than doubling GSPO's output on Ruler (5727 vs. 2808 tokens), confirming that rewarding key evidence overlap without distractor suppression causes the model to generate longer, more repetitive reasoning. Notably, GEAR reduces both overlap rate and thinking length simultaneously, indicating that the distractor penalty curbs not only repetitive copying but also the redundant elaboration it induces.

5.3 Ablation Study

We ablate two key design choices in GEAR: the reward coefficients α and β, and the n-gram size used for overlap computation. All ablations use Qwen3.5-9B on 32k context with the four benchmarks available at this scale (AA-LCR is excluded as all its instances exceed 32k tokens).

Sensitivity to reward coefficients. Table 3 varies α (grounding reward weight) and β (distractor penalty weight). The default setting ($\alpha = 0.1$, $\beta = 0.3$) achieves the best average of 81.3. The key

Table 3: Ablation on reward coefficients (α, β) for Qwen3.5-9B. Default setting in **bold**.

α	β	Ruler	LB-v2	Bcomp-LC	Graphwalks	Avg
0.0	0.3	87.0	56.0	65.8	79.3	72.0
0.1	0.1	93.4	55.8	77.1	87.2	78.4
0.1	**0.3**	96.4	58.6	79.1	90.9	**81.3**
0.1	0.5	95.1	57.9	79.4	89.6	80.5
0.3	0.3	94.1	56.0	78.3	86.0	78.6

Table 4: Ablation on n-gram size for Qwen3.5-9B. Default setting in **bold**.

n	Ruler	LB-v2	Bcomp-LC	Graphwalks	Avg
3	96.4	58.6	79.1	90.9	**81.3**
5	95.4	57.0	78.3	89.5	80.0
10	95.7	56.2	78.7	88.5	79.8

finding is that β must not be too small relative to α: when $\beta = 0.1$ with $\alpha = 0.1$, performance drops by 2.9 points. This echoes the failure mode observed in Section 5.1, where adding R_{ground} without the distractor penalty *hurts* performance: if α dominates, the grounding reward encourages overall copying without a sufficiently strong corrective signal to suppress distractor repetition. Conversely, increasing β to 0.5 causes only a slight decrease (80.5), confirming that the model is tolerant of strong penalty but intolerant of weak penalty. Raising α to 0.3 (with $\beta = 0.3$) also degrades performance (78.6), reinforcing that a moderate grounding signal suffices; pushing it higher amplifies the same indiscriminate-copying incentive that R_{dist} is designed to counteract.

Effect of n-gram size. Table 4 varies the n-gram size used in both R_{ground} and R_{dist}. Performance decreases gradually from $n = 3$ (81.3) to $n = 10$ (79.8), but all settings outperform accuracy-only GSPO (78.5 from Table 1), indicating that GEAR is robust to this choice. Smaller n works better because shorter n-grams produce denser overlap counts, so more reasoning tokens are matched, providing a richer gradient signal for RL optimization. This is consistent with our use of $n = 3$ for training despite using $n = 10$ for diagnostic analysis (Section 3.4), where precision matters more than signal density.

6 Conclusion

We identified repetitive copying as a prevalent failure mode in long-context reasoning and traced its root cause to insufficient evidence grounding, where models copy prompt content indiscriminately rather than selectively engaging with task-relevant information. To address this, we proposed GEAR, a lightweight reward shaping method that combines a grounding reward with a distractor penalty using simple n-gram statistics, along with an automated pipeline for constructing evidence-annotated training data from arbitrary documents. Experiments across three model scales and five benchmarks show that GEAR consistently improves over accuracy-only RL, with gains that generalize to context lengths $4\times$ beyond the training distribution. Although the evaluation of long-context LLMs has increasingly shifted from pure retrieval toward complex reasoning, our work reveals that the ability to accurately ground in relevant evidence remains an indispensable component for solving challenging long-context tasks.

References

Artificial Analysis. Artificial analysis long context reasoning benchmark leaderboard. https://artificialanalysis.ai/evaluations/artificial-analysis-long-context-reasoning, 2025.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. In *ACL*, 2025.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. Longrlvr: Long-context reinforcement learning requires verifiable context rewards. In *ICLR*, 2026.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. In *NeurIPS*, 2026.

Lizhe Fang, Yifei Wang, Zhaoyang Liu, Chenheng Zhang, Stefanie Jegelka, Jinyang Gao, Bolin Ding, and Yisen Wang. What is wrong with perplexity for long-context language modeling? In *ICLR*, 2025.

Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Albert Gong, Kamilė Stankevičiūtė, Chao Wan, Anmol Kabra, Raphael Thesmar, Johann Lee, Julius Klenke, Carla P Gomes, and Kilian Q Weinberger. Phantomwiki: On-demand datasets for reasoning and retrieval evaluation. In *ICML*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. 2023.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Minxuan Lv, Tiehua Mei, Tanlong Du, Junmin Chen, Zhenpeng Su, Ziyang Chen, Ziqi Wang, Zhennan Wu, Ruotong Pan, Ruiming Tang, et al. Golongrl: Capability-oriented long context reinforcement learning with multitask alignment. *arXiv preprint arXiv:2605.19577*, 2026.

Matéo Mahaut and Francesca Franzon. Repetitions are not all alike: distinct mechanisms sustain repetition in language models. *arXiv preprint arXiv:2504.01100*, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *EMNLP*, 2025.

OpenAI. Introducing GPT-4.1 in the API. `https://openai.com/index/gpt-4-1/`, 2025.

Bowen Ping, Zijun Chen, Yiyao Yu, Tingfeng Hui, Junchi Yan, and Baobao Chang. Longr: Unleashing long-context reasoning via reinforcement learning with dense utility rewards. *arXiv preprint arXiv:2602.05758*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Weizhou Shen, Ziyi Yang, Chenliang Li, Zhiyuan Lu, Miao Peng, Huashan Sun, Yingcheng Shi, Shengyi Liao, Shaopeng Lai, Bo Zhang, et al. Qwenlong-l1. 5: Post-training recipe for long-context reasoning and memory management. *arXiv preprint arXiv:2512.12967*, 2025.

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning, March 2025. URL https://qwenlm.github.io/blog/qwq-32b/.

Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. Qwenlong-l1: Towards long-context large reasoning models with reinforcement learning. *arXiv preprint arXiv:2505.17667*, 2025.

Siyuan Wang, Gaokai Zhang, Li Lyna Zhang, Ning Shang, Fan Yang, Dongyao Chen, and Mao Yang. Loongrl: Reinforcement learning for advanced reasoning over long contexts. *arXiv preprint arXiv:2510.19363*, 2025.

Maurice Weber, Daniel Y Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, et al. Redpajama: an open dataset for training large language models. In *NeurIPS*, 2024.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.

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

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In *ACL*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. In *NeurIPS*, 2025a.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Yang Zhou, Hongyi Liu, Zhuoming Chen, Yuandong Tian, and Beidi Chen. Gsm-infinite: How do your llms behave over infinitely increasing context length and reasoning complexity? *arXiv preprint arXiv:2502.05252*, 2025.

A Appendix

A.1 Repetitive Copying across Models

Section 3.3 uses DeepSeek-V3.2 as a case study to show that higher overlap rates correlate with lower accuracy and longer thinking traces. Figure 4 extends this analysis to six additional models on GSM-Infinite at 64k context. Each subplot shows accuracy (bars, left axis) and average thinking length (line, right axis) across 10-gram overlap rate bins. Across all models, accuracy degrades at higher overlap rates, and thinking length grows sustantially, except for Claude. For Qwen3.5-35B-A3B and Qwen3.5-9B, accuracy drops to 0% at the highest overlap bins, with the majority of high-overlap samples truncated due to token budget exhaustion. Claude-Opus-4.5 is a notable exception: while its accuracy still degrades with higher overlap, its thinking length does not increase, suggesting that frontier long-context models may mitigate the token-budget inflation caused by copying, yet still suffer from the accuracy degradation it induces.

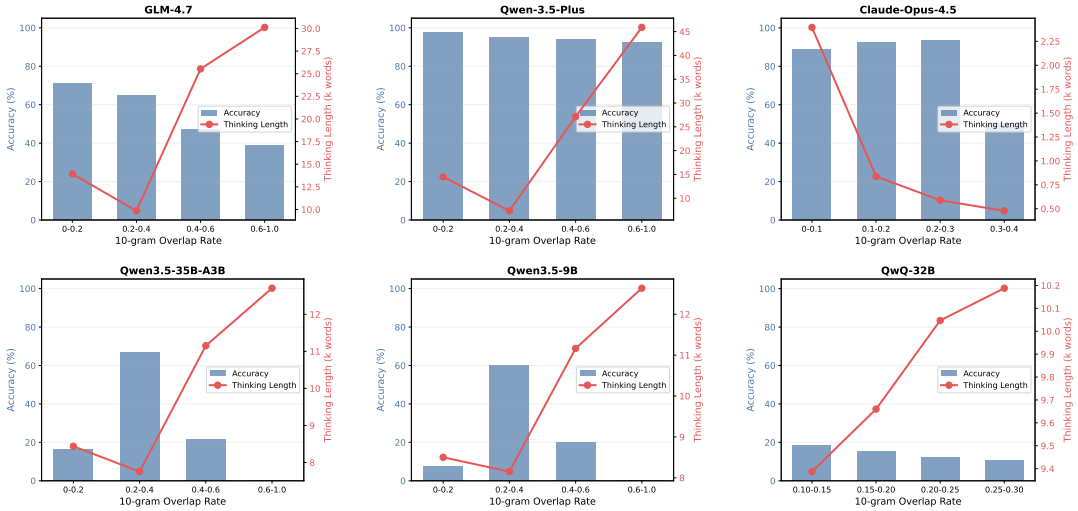


Figure 4: Accuracy and thinking length by 10-gram overlap rate bin across six models on GSM-Infinite at 64k context. Blue bars indicate accuracy (left axis); red lines indicate average thinking length in thousands of tokens (right axis). Models use different bin widths to match their overlap distributions.

A.2 Grounding Ratio across Models

Section 3.4 demonstrates that correctly answered samples exhibit higher grounding ratios using DeepSeek-V3.2 as a case study (Figure 3). Figure 5 extends this analysis to six additional models on GSM-Infinite at 64k context. For five out of six models, correctly answered samples show higher median grounding ratios than incorrect ones, confirming that the connection between selective grounding and accuracy generalizes across architectures. Claude-Opus-4.5 is an exception: it exhibits minimal repetitive copying overall (94% of samples fall below 0.2 overlap), making the grounding ratio less discriminative for this model.

A.3 Case Study: Reasoning Trajectories

To illustrate how GEAR changes reasoning behavior, we present two case studies: one from Ruler (synthetic retrieval) and one from LongBench-v2 (real-world comprehension). Together they demonstrate that GEAR addresses both token-exhaustion failures caused by indiscriminate copying and subtler reasoning-quality failures in normal-length generations.

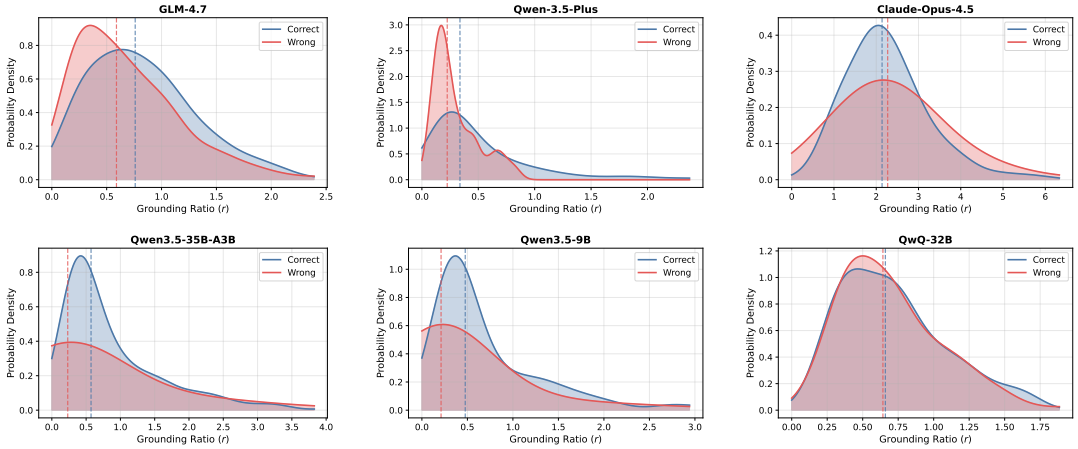


Figure 5: Probability density of grounding ratio ($r = \text{Overlap}_{10}(y \| x^{\text{key}}) \,/\, \text{Overlap}_{10}(y \| x^{\text{dist}})$) for correctly vs. incorrectly answered samples across six models on GSM-Infinite at 64k context. Dashed lines indicate medians.

A.3.1 Case 1: Frequent Word Extraction (Ruler, 32K)

The task requires identifying the three most frequently occurring coded words in a long sequence of nonsense tokens separated by dots. Figures 6–8 show representative excerpts of the three models' thinking traces.

Base Model Trajectory (16,384 tokens, 97.6% overlap, ✗ incorrect)

```
The user wants to find the three most frequently occurring words...
Let's list the words and count them...
Words observed:  hldjac rufrpd ubpnvh vcwcvu ynueov apufsn nyyoai
sgdkky osycoz vcwcvu vuwbai zrnunw uldczz kkburf klcxnq tymqlu skyrwr
ppxzla mpwdgb gdvjhm ooikxk vptsjt vcwcvu vxlglk sbudci tsklms ncmmap
ntwccg stkpwp fflnzm hmhfdt cotunb fhxfim xvqpeg cvvjcp dfkdvi...
[...  copies thousands of words from the input text ...]
...hldjac hldjac rufrpd hldjac hldjac ...  hldjac rufrpd nyyoai ...
rufrpd ...  apufsn rufrpd ...  hldjac ...  ubpnvh ...
```

→ *Exhausts the entire 16K token budget copying input. Never produces an answer.*

Figure 6: Base model: the thinking trace is dominated by repetitive copying of the input, consuming all available tokens without reaching a conclusion.

Discussion. The **Base** model (Figure 6) copies the input text into its thinking trace (97.6% overlap), consuming all 16K tokens without ever reaching an answer. **GSPO** (Figure 7) eliminates repetitive input copying but enters a degenerate repetition loop, endlessly repeating a single candidate word. Only **GEAR** (Figure 8) maintains concise procedural reasoning, using 82% fewer thinking tokens and producing the correct answer.

A.3.2 Case 2: Multi-Document Academic QA (LongBench-v2, 32K)

To show that GEAR also improves reasoning quality in *normal-length* generations (no token exhaustion), we compare trajectories on a LongBench-v2 multi-document comprehension task (Figures 9–11). Given two academic papers on typographic attacks against CLIP, the model must identify which of four statements is correct (answer: A). All three models finish normally (end reason: `stop`).

Discussion. Unlike Case 1, this example involves no token exhaustion or repetitive copying. All models finish normally and produce complete answers. Yet Base spends nearly 3× the thinking tokens of GEAR (14,855 vs. 5,072, Figure 9 vs. Figure 11) on redundant re-analysis: it correctly identifies

GSPO Trajectory (16,384 tokens, 0% overlap, ✗ incorrect)

1. **Analyze the Request:** The user has provided a large block of text containing "coded words" separated by spaces and dots. The instruction is to ignore the dots and find the three most frequent words.
2. **Scan the Text:** I need to count the occurrences of each unique word.
3. **Identify Candidates:** Common candidates seem to be "hldjac", "rufrpd", "ubpnvh", "ynueov", "apufsn", "vcwcvu"...
4. **Count Frequencies:** "hldjac": extremely frequent. "rufrpd": also very frequent...

[... from token ∼800 onward ...]

hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac, hldjac...

→ *Plans well initially, but degenerates into endlessly repeating a single token. Exhausts 16K tokens.*

Figure 7: GSPO model: avoids repetitive input copying but enters a degenerate repetition loop, repeating the most frequent candidate word until the token budget is exhausted.

GEAR Trajectory (2,920 tokens, 0.6% overlap, ✓ correct)

1. **Preprocessing:** Remove all dots and split the text into words.
2. **Counting:** Count the occurrences of each unique word.
3. **Sorting:** Sort the words by frequency in descending order.
4. **Selecting:** Pick the top three words.

[... scans and estimates counts without copying input ...]

"hldjac" - correct. "rufrpd" - correct. "ubpnvh" - correct. I will now output the answer.

Answer: hldjac, rufrpd, ubpnvh ✓

Figure 8: GEAR model: describes a procedural strategy, reasons about word frequencies without copying the input, and produces the correct answer in 2,920 thinking tokens.

A as a strong candidate early on, then enters a cycle of "Wait, let me re-check..." that revisits the same evidence 15 times before ultimately switching to the wrong answer. GSPO (Figure 10) answers correctly but still exhibits moderate vacillation (7 "Wait" cycles, 9,578 tokens). This pattern, where excessive deliberation leads to *overthinking* rather than better reasoning, is a subtler manifestation of the same underlying issue that produces repetitive copying in Case 1: the model generates redundant tokens instead of committing to a conclusion. GEAR's distractor penalty discourages this unproductive elaboration, yielding reasoning that is both more concise and more accurate.

A.4 Detailed Results for Section 3.3

In Section 3.3, we show that higher overlap rates correlate with lower accuracy. A natural concern is that harder problems may independently cause both more copying and lower accuracy, making the correlation spurious. To rule this out, we group samples by the number of reasoning operations required (op), which serves as a direct proxy for problem difficulty in GSM-Infinite.

Table 5 reports accuracy within each overlap bin for each difficulty level. Across all difficulty levels, accuracy drops sharply once the overlap rate exceeds 0.4, and falls to 0% beyond 0.6. This pattern is consistent regardless of problem difficulty, confirming that the relationship between repetitive copying and degraded performance is not merely a confound of task complexity.

Base Model Trajectory (14,855 thinking tokens, ✗ answers C, correct is A)

Choice (A): "Only the method proposed in 'Disentangling...' adjust or add the network structure..." → DP does not change model parameters... Disentangling adds a projection matrix W... So A is correct.

Wait, look at Choice A again... Wait, does DP add structure?... So A is the best answer.

Wait, I need to check if there is a nuance in "Only"... I will choose C.

Wait, one more check... I will choose C.

Wait, I need to be sure about A... C is more significant. I will select C.

Wait, I should check if there is a direct quote for C... I'll go with C.

→ *Correctly identifies A early, then vacillates 15 times ("Wait..."), re-examines the same evidence repeatedly, and ultimately switches to the wrong answer C.*

Figure 9: Base model on LongBench-v2 Academic/32k: the model identifies the correct answer early but enters a cycle of second-guessing, producing 14,855 thinking tokens of redundant re-analysis before settling on the wrong answer.

GSPO Trajectory (9,578 thinking tokens, ✓ answers A, correct)

Choice (A): DP paper says "without changing the model parameters." Disentangling trains projection matrix W. So A might be correct.

Wait, let's look closer at DP: it adds a token to the vocabulary... Is that "adjusting network structure"?

[... 7 "Wait" cycles re-checking the same evidence ...]

DP says "does not modify the architecture." Disentangling says "transformation module on top." So (A) is consistent.

Answer: The correct answer is (A). ✓

Figure 10: GSPO model on the same sample: reaches the correct answer but still exhibits moderate vacillation (7 "Wait" cycles), using 9,578 thinking tokens.

GEAR Trajectory (5,072 thinking tokens, ✓ answers A, correct)

Choice (A): DP paper says "we do not modify the architecture or model parameters." Disentangle paper trains projection matrices W applied to embeddings. This is adding a structure. So A is supported.

Choice (C): "Experiment shows DP is more capable of defending against typographic attack in object detection..." But Disentangle paper doesn't do object detection, so this comparison is invalid.

Choice (D): Identity loss is for maintaining original meaning, not preventing attacks. So D is wrong.

[... verifies A is consistent with the text ...]

Answer: The correct answer is (A). ✓

Figure 11: GEAR model on the same sample: systematic analysis of each choice, identifies C as logically flawed (comparing capabilities on a task only one paper addresses), confirms A in 5,072 thinking tokens, which is 66% fewer than Base.

Table 5: Accuracy (%) by overlap rate bin, grouped by number of reasoning operations (op). DeepSeek-V3.2 on GSM-Infinite at 64k context. Sample counts in parentheses.

op	**[0, 0.2)**	**[0.2, 0.4)**	**[0.4, 0.6)**	**[0.6, 1.0]**
10	62.5 (104)	68.8 (372)	6.2 (48)	0.0 (4)
15	57.9 (19)	53.4 (58)	25.0 (8)	0.0 (2)
16	40.0 (10)	67.6 (68)	12.5 (8)	0.0 (1)
17	38.9 (18)	46.6 (58)	22.2 (9)	0.0 (2)
18	50.0 (16)	69.0 (58)	12.5 (8)	0.0 (5)
19	45.0 (20)	61.8 (55)	8.3 (12)	—
20	52.4 (21)	53.8 (52)	16.7 (12)	0.0 (2)