Tübingen-CL at SemEval-2026 Task 12: Reinforcement Learning and Verification for Abductive Reasoning

Bolun Liang
bolun.liang@student.uni-tuebingen.de

Ayperi Khudaybergenova
ayperiayperi23@gmail.com

Shashikala Kankanamge
shashikala.alawaththa-kankanamge@student.uni-tuebingen.de

Abstract

We investigate the reliability of verifier-based pipelines for abductive reasoning in SemEval-2026 Task 12. While reinforcement learning improves the base generator's performance, we find that incorporating a small-model verifier introduces a significant generalization gap: although effective on validation data, the verifier systematically degrades correct predictions on the unseen test set by appending false positives. Furthermore, we reveal a critical vulnerability in the official evaluation metric, which assigns zero reward to abstentions but does not sufficiently penalize incorrect selections. This asymmetry enables trivial heuristic strategies such as blindly selecting a default option to substantially inflate performance, even outperforming more principled reasoning systems. Our analysis demonstrates that current evaluation protocols can misrepresent true reasoning ability and highlights the need for more robust verification methods and scoring schemes.

1 Introduction

Abductive reasoning requires identifying the most plausible causes of partially observed events. In SemEval-2026 Task 12 (Cao et al., 2026), systems must select all and only the correct causes from a set of candidates based on an English-language dataset, a task that remains challenging for large language models under noisy evidence.

Recent approaches employ generation verification pipelines, where a verifier refines model outputs. However, we show that small-model verifiers exhibit a significant generalization gap: while improving validation performance, they degrade correct predictions on unseen test data by introducing false positives.

We further identify a critical flaw in the evaluation metric, which assigns zero reward to abstentions but insufficiently penalizes incorrect selections. This asymmetry allows trivial heuristics such as always selecting a default option to substantially inflate performance. In high-stakes real-world applications, however, a safe abstention under insufficient evidence is inherently more valuable than hallucinating false causal links. Our analysis highlights limitations in both verification strategies and current evaluation protocols. To ensure reproducibility, our complete codebase and configurations are publicly available at https://github.com/AlanLoeng/SemEval2026-Task12-RL-Verifier.

2 Related Work

Prior work has explored abductive and causal reasoning with large language models.(Shi et al., 2023) show that few-shot prompting can elicit plausible explanations for event prediction, demonstrating the effectiveness of in-context reasoning without additional training.

However, prior studies show that large language models often struggle with reliable reasoning in the absence of structured prompting (Wei et al., 2023). This motivates approaches that go beyond prompting and incorporate feedback-based optimization.

Causal reasoning in multi-document settings has also been studied through contextual integration methods (Wang et al., 2025), where structured evidence selection improves explanation quality.

In contrast, we optimize reasoning behavior using reinforcement learning and integrate a verification mechanism to improve robustness under noisy or competing evidence.

3 Data Setup & Preprocessing

3.1 Validation Set

Because the officially provided development and sample sets (Cao et al., 2026) are true subsets of the 36-topic training data, we constructed an independent validation set to prevent data leakage. Specifically, this validation split is dedicated exclusively to hyperparameter tuning, early stopping,

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2621–2629

and ablation studies, reserving the official test set for final unbiased evaluation. To maintain the thematic diversity of the original corpus, we implemented a stratified sampling approach. We first utilized Qwen3-30B (Yang et al., 2025) to categorize all training topics into predefined domains. We then proportionally sampled 6 topics to form our validation set: two from Politics (IDs 7 and 27), and one each from Economy (ID 29), Technology (ID 18), Science and Public Health (ID 2), and Environment (ID 33). This split yielded a robust 6-topic validation set, leaving the remaining 30 topics strictly for training. The ground-truth labels for this validation set are the original annotations provided in the official training release. By mirroring the domain distribution of the full 36-topic corpus, this split provides a representative in-distribution benchmark for local tuning. However, as our subsequent analysis reveals, while this in-distribution evaluation was highly effective for tuning the base generative model, it inherently could not expose the decoupled verifier's severe generalization gap prior to the final unseen test.

3.2 Context Management Pipeline

Applying Group Relative Policy Optimization (GRPO) requires generating multiple simultaneous reasoning trajectories, which imposes strict memory constraints on the context window. To accommodate extensive reasoning traces, we implemented a symmetric 16,384-token allocation strategy, reserving 8,192 tokens for input evidence and 8,192 for the generator's output.

To fit massive raw retrieved documents within this strict limit without losing causally pivotal information, we designed a two-stage data refinement pipeline. First, an instruction-tuned LLM performs soft filtering, explicitly evaluating and retaining only documents with direct or indirect causal relevance to the target event. Second, a cross-encoder reranks the filtered documents by semantic relevance. This architecture ensures the most critical evidence resides at the top of the context window, enabling safe hard-truncation at the strict boundary (see Section 4.6 for specific compression statistics).

4 Method

4.1 Base Models

Qwen We use **Qwen3-4B-Thinking-2507** (Yang et al., 2025) as our primary generator. Its strong chain-of-thought capabilities serve as the foundation for our generation-verification architecture.

Ministral We incorporate **Ministral-3-8B-Reasoning-2512** (Liu et al., 2026) to introduce architectural diversity into our system. As a larger, fundamentally different model family, it validates the generalizability of our GRPO training and provides complementary reasoning patterns for the joint-encoding ensemble.

Microsoft DeBERTa-v3-base For the discriminative verification module, we employ **Microsoft DeBERTa-v3-base** (He et al., 2023). Operating as a sequence classifier, it leverages its disentangled cross-attention mechanism to explicitly assess the logical coherence of abductively generated explanations.

4.2 Reinforcement Learning On Qwen & Ministral

To improve abductive reasoning without relying on large-scale annotated chain-of-thought data, we adopt a Reinforcement Learning (RL) approach instead of conventional Supervised Fine-Tuning (SFT). Specifically, we employ Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a memory-efficient strategy designed for reasoning-focused training.

Unlike PPO, GRPO eliminates the need for a separate value network by using the mean reward of the sampled group as the baseline, which allows for training without a computationally expensive auxiliary model. To operate within hardware constraints while preserving a large context window, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2021) for parameter-efficient fine-tuning. For each training prompt, the policy is optimized using relative reward differences within the sampled group. To explore architectural generalization, we applied this configuration to both Qwen and Ministral models, adjusting only the native chat templates for parallel training (see Section 4.6 for granular training hyperparameters and hardware configurations).

4.3 Reward Design

One of the major challenge in applying RL to an event reasoning task is preventing the model from generating either overly brief guesses or endlessly verbose, rambling chain-of-thought traces. To address this, we designed a composite reward system consisting of three distinct functions: Format Reward, Score Reward, and a novel Structural Length Reward.

Format Reward Before evaluating the logic, the system must parse the output. Requiring JSON is to allow the evaluation script to stably parse and score, and to extract the reasoning field as structured input for the subsequent DeBERTa. The block is an exclusive marker used by reasoning models to isolate the internal Chain-of-Thought (CoT) brainstorming process. This rule-based reward grants a flat +0.5 if a valid JSON object can be successfully extracted from the completion, effectively ignoring any generated tags or markdown wrappers. If parsing fails, the reward is 0.0, strictly enforcing the required output structure.

Score Reward To provide a more pronounced optimization signal during GRPO training, we derive the **Score Reward** by applying a scaling factor of 2.0 to the per-instance **Score** (defined in Section 5.1). This ensures that the policy receives a substantial positive reinforcement (+2.0) for an Exact Match, +1.0 for a Partial Match, and 0.0 for hallucinations or conservative abstentions.

Structural Length Reward To explicitly guide reasoning verbosity and prevent fatal Out-Of-Memory (OOM) hard truncations, we introduced a two-part length reward with a combined range of $[-0.8, +0.8]$. This reward acts as a strict structural regularizer rather than penalizing necessary logic. We define L_{think} as the token length of the internal <think> process, and L_{json} as the combined token length of the final justifications. The component rewards are computed as follows:

$$R_{\text{think}} = \begin{cases} 0.4, & \text{if } L_{\text{think}} \leq 2048 \\ 0.4 \cdot \frac{4096 - L_{\text{think}}}{2048}, & \text{if } 2048 < L_{\text{think}} \leq 4096 \\ -0.4 \cdot \frac{L_{\text{think}} - 4096}{4096}, & \text{if } 4096 < L_{\text{think}} \leq 8192 \\ -0.4, & \text{if } L_{\text{think}} > 8192 \end{cases}$$

$$R_{\text{json}} = \begin{cases} 0.4 - 0.8 \cdot \frac{|L_{\text{json}} - 400|}{200}, & \text{if } 200 \leq L_{\text{json}} \leq 600 \\ -0.4, & \text{otherwise} \end{cases}$$

This mathematical formulation elegantly forces the model to distill its lengthy internal brainstorming into sharp, high-density explanations bounded around 400 tokens, while strictly applying linear penalties to runaway reasoning loops that approach the 8,192-token context limit.

4.4 Generation-Verification Architecture

Despite improvements from Reinforcement Learning, generative models remain prone to overconfidence and hallucinated causal links under strong distractors. To address this, we adopt a Generation–Verification pipeline that pairs an RL-tuned generator with a discriminative encoder-based verifier. The core intuition is that verifying a proposed explanation is easier and more reliable than generating one.

In the Generation phase, the model produces structured predictions for all options. For each option, we extract the binary decision and its concise justification. We explicitly discard the internal <think> traces and retain only the final reasoning, ensuring that the verifier evaluates distilled conclusions rather than noisy intermediate chains of thought.

In the Verification phase, each option is evaluated independently as a binary classification task. The input sequence concatenates the target event, candidate option, generator prediction and rationale, and reference document titles.

We restrict the verifier input to document titles rather than full texts. Due to the 1024-token limit, including full documents introduces severe truncation and low-signal noise, which empirically degrades performance. Titles preserve high-level causal semantics while enabling stable and consistent encoding.

The verifier is trained using naturally occurring positive and negative examples derived from candidate options and ground-truth labels, without artificial negative oversampling. This ensures that the classification task reflects the original decision structure of the dataset.

The final prediction consists of all options assigned a positive label by the verifier (see Section 4.6 for implementation details and Appendix A.3 for the input template).

4.5 Ensemble Strategy

To further improve robustness against single-model hallucination patterns, we integrated an ensemble strategy that performs joint contextual verification using both Qwen and Ministral generators. Instead of a heuristic voting mechanism, the structured predictions and distilled rationales from both models are concatenated into a unified input sequence alongside the target event and reference titles. The DeBERTa verifier then processes this combined representation, leveraging its cross-attention mechanism to simultaneously weigh the potentially competing logics before making a final binary classification. However, likely due to the absence of a third generator to resolve conflicting reasoning

patterns, this dual-model joint encoding ultimately underperforms the standalone Gen-Verif pipeline. Consequently, the single-generator architecture remains the primary focus of our subsequent analysis.

4.6 Implementation Details

To ensure reproducibility, our system's hardware and software configurations are standardized as follows.

Context and Retrieval The input pipeline manages an average of 36,070 raw tokens per topic. We employ a symmetric 16,384-token window, with 8,192 tokens reserved for input documents and 8,192 for generation. Document refinement is conducted via a two-stage pipeline:

- **LLM Soft Filtering (LLMSF):** Utilizing **Qwen3-30B-Instruct-2507** (Yang et al., 2025) to prune the document count from an average of 22.9 to 14.7 per topic.
- **Semantic Reranking:** Utilizing **BAAI/bge-reranker-v2-m3** (Li et al., 2023; Chen et al., 2024) with the target event description as the query. Documents are capped at 5,000 characters each; the top 8–10 reranked documents are prioritized to fit the 8,192-token input ceiling.

RL Training GRPO training utilizes the Unsloth library (Han et al., 2023) on 2 × NVIDIA A100 (80GB) GPUs in `bfloat16` precision. We apply Low-Rank Adaptation (LoRA) with rank $r = 64$ and $\alpha = 64$ to all linear layers within the attention and feed-forward modules. The sampling group size is set to $G = 4$ per prompt. The Qwen3-4B model converged at 492 steps with a learning rate of 5e-7.

Verifier Configuration The **Microsoft DeBERTa-v3-base** verifier (He et al., 2023) is implemented as a binary sequence classifier with a maximum sequence length of 1,024 tokens. The training set was derived from the Qwen generator's inference on the 30-topic training split (see Section 3.1). Positive and negative training samples were naturally formed by evaluating all four candidate options per event against their ground-truth labels, without utilizing artificial negative oversampling. Following the template in Appendix A.3, the input sequence concatenates the target event, candidate option, generator's predicted verdict and reasoning, and reference document titles via the `[SEP]` token. The model was trained for up to 6 epochs on an NVIDIA GTX 1080Ti with a total batch size of 32, a learning rate of 2e-5, and dropout rates of 0.2. The training process utilized an early stopping strategy and converged at 1.32 epochs.

5 Experiments & Analysis

5.1 Evaluation Metric

The official performance metric for SemEval-2026 Task 12 (Cao et al., 2026) is the **Score**, calculated as the mean of per-instance scores across the dataset. For each instance, a score of 1.0 is awarded for an **Exact Match** (identifying all and only correct options). A **Partial Match**—defined as a non-empty proper subset of the ground truth with no incorrect options—yields 0.5. Any incorrect selection or empty prediction (abstention) results in a **Mismatch** with a score of 0.0.

5.2 Results

Model	Val.	Test (True)	Test (+FB)
Qwen3-32B†	0.4518	0.7059	0.7296
R1-Distill (70B)†	0.5241	0.5449	0.5596
Ministral (RL)	0.6406	0.5547	0.5997
Qwen (RL)	0.6988	**0.7239**	**0.7835**
Gen-Verif	**0.7209**	0.7149	0.7631
Ensemble	0.7150	0.6977	0.7312

†*Evaluated as zero-shot baselines configured according to their official deployment best practices. See Appendix A.1 for prompts.*

Table 1: System score on the validation and test sets. '+FB' denotes the heuristic Fallback (Guess A) applied to empty predictions.

Experimental results (Table 1) highlight the effectiveness of our RL-tuned 4B generator, which outscores both the heavily parameterized Qwen3-32B (Yang et al., 2025) (0.7059) and DeepSeek-R1-Distill-Llama-70B (Guo et al., 2025) (0.5449) zero-shot baselines on the test set. However, extending the system with a verifier introduced a distinct generalization gap. While Gen-Verif achieved the peak validation score (0.7209), the standalone Qwen model outperformed it on the official test set (0.7239 vs. 0.7149). The DeBERTa verifier inadvertently degraded robust predictions on the test distribution. The ensemble approach (0.6977) similarly underperformed.

Furthermore, we expose a critical vulnerability in the official evaluation metric, which strictly assigns 0.0 to empty predictions (abstentions). We

applied a naive fallback (**+FB**): forcefully overriding any empty prediction with a blind guess of Option 'A'. This simple heuristic artificially inflated Qwen's test score from 0.7239 to 0.7835 directly accounting for our official 0.78 leaderboard submission. This reveals that the current metric inadvertently incentivizes reckless guessing over honest abstention. Without an asymmetric penalty for false positives, simple heuristics can exploit the evaluation, thereby obscuring the true limits of a model's causal reasoning.

5.3 Ablation For Filtering & Reranking Pipeline

We evaluated four stages of evidence refinement to investigate the impact of context management. Results are detailed in Table 2.

Configuration	Gen. (RL Qwen)	Gen-Verif
Raw (Unprocessed)	0.3112	0.3655
LLMSF Only	0.5823	0.6004
Reranker Only	0.6667	0.6868
LLMSF + Reranker	**0.6988**	**0.7209**

Table 2: Ablation of the context management pipeline.

Impact of Raw Context and Truncation The *Raw* configuration yields the lowest score (0.3112), illustrating the detrimental impact of low-quality context. Without prioritization, the 8,192-token ceiling frequently truncates causally pivotal evidence, leaving the generator to reason with incomplete information.

Pruning vs. Prioritization While LLM-based filtering (*LLMSF Only*) removes noise to reach 0.5823, it is outperformed by the *Reranker Only* strategy (0.6667). This suggests that for abductive reasoning, ensuring semantically relevant evidence resides at the beginning of the context window is more critical than a broad reduction of thematic noise.

Synergy in the Optimal Pipeline The *Optimal* configuration (0.6988) proves that synergy is essential: soft filtering prunes irrelevant documents to expand the effective window, while reranking maximizes causal density within that space.

5.4 Reward Function Ablation

To analyze the influence of structural regularization, we ablate the composite reward function components. Table 3 shows the overall score degradation when the **Format Reward (FMR)** or the **Structural Length Reward (SLR)** is removed. Beyond overall scores, we introduce an error taxonomy to diagnose behavioral shifts: Over-selection (Type A), Under-selection (Type B), Incorrect Selection (Type C1), Abstention (Type C2, empty prediction), and Format Failure (Type D). Table 4 details the distribution of these errors across ablated configurations.

Reward Configuration	Score
Optimal (Full Composite Reward)	**0.6988**
w/o SLR	0.6767
w/o FMR	0.6687

Table 3: Ablation of the composite reward components.

Category	Optimal	w/o SLR	w/o FMR
Exact Match	157 (63.05%)	152 (61.04%)	152 (61.04%)
Type A (Over)	7 (2.81%)	5 (2.01%)	8 (3.21%)
Type B (Under)	34 (13.65%)	33 (13.25%)	29 (11.65%)
Type C1 (Incorrect)	19 (7.63%)	14 (5.62%)	11 (4.42%)
Type C2 (Abstention)	30 (12.05%)	42 (16.87%)	47 (18.88%)
Type C1+C2	**49 (19.68%)**	**56 (22.49%)**	**58 (23.30%)**
Type D (Format Fail)	2 (0.80%)	3 (1.20%)	2 (0.80%)

Table 4: Error distribution across ablated reward policies ($N = 249$).

The empirical data validates FMR and SLR as essential heuristic regularizers. As shown in Table 4, removing FMR causes Type C2 to surge from 30 to 47. Although Type C1 appears to decrease, the aggregate complete failure rate (C1+C2) rises markedly to 23.30%. Similarly, the removal of SLR increases the C1+C2 total to 22.49%. These ablation results yield a straightforward empirical conclusion: the presence of these structural rewards is strictly necessary to maintain a functional reasoning policy. Without them, the generator collapses into a conservative "abstention bottleneck," yielding empty prediction sets rather than attempting valid logical deductions.

5.5 Results & Error Analysis

The Generalization Gap As shown in Table 5, the verifier exhibits a strictly unidirectional behavior: it exclusively appends options to the base generator's predictions without ever removing any. On the validation set, this additive mechanism yielded a favorable trade-off. However, because the pipeline's hyperparameters and early stopping criteria were optimized for this validation distribution,

State Transition (Qwen → Gen-Verif)	Val.	Test
Score Degradation (-1.0)		
Exact Match → Type A (Over)	1	22
Score Neutral (0.0)		
Type C1 (Incorrect) → Type A (Over)	7	20
Type C2/D → Type A/C1/C2	0	8
Partial Recovery (+0.5)		
Type C2 (Abstention) → Type B (Under)	10	19
Type B (Under) → Exact Match	1	2
Full Recovery (+1.0)		
Type C2 (Abstention) → Exact Match	0	6
Type D (Format) → Exact Match	2	0

Table 5: State transitions from the base generator to the verification pipeline. Error types are formally defined in Section 5.4. The verifier systematically degraded 22 perfect predictions into Type A errors on the test set.

this performance failed to generalize to the test set. Instead, the verifier systematically appended false positive relations to otherwise perfect predictions, degrading 22 Exact Matches into Type A (Over) errors. Since the official evaluation metric strictly assigns 0.0 to Type A errors, this systematic error compounding entirely accounts for the overall performance drop of the Gen-Verif pipeline.

The Metric Exploit Furthermore, our error analysis exposes a critical vulnerability in the evaluation metric. On the test set, Qwen produced 92 empty predictions (honest abstentions, Type C2) and 6 format failures (Type D). By bypassing the verifier and applying the **+FB** heuristic blindly overriding these 98 unresolved instances with Option 'A' the system successfully exploited the metric's asymmetric penalization. Specifically, this blind guessing converted the 98 empty/failed predictions into **30 Exact Matches (+1.0)** and **13 Partial Matches (+0.5)**. This statistical evidence confirms that the current metric inadvertently incentivizes hard-coded guessing over calibrated abstention.

Case Study: Verifier Over-selection To concretely illustrate this generalization failure, consider a test instance where the base generator succeeded but the verifier appended a false positive:

Event: Chang'e 5 launched from the lunar surface Dec. 3.
[A] (Ground Truth & Qwen): Chang'e 5 successfully landed in the Ocean of Storms region on the moon near Mons Rümker on Dec. 1.
[B]: Chang'e-5 raised a Chinese flag on the Moon.
[C]: Chang'e 5 mission entered lunar orbit on November 28.
[D] (Appended by Verifier): Chang'e 5 launched on November 23 atop a Long March 5 rocket.

While Qwen correctly isolated the singular direct cause (Option A), the DeBERTa verifier was misled by the spurious semantic associations in Option D. Lacking the generative model's deep causal chain-of-thought, the sequence classifier treated strong lexical co-occurrences (e.g., connecting "launched" with "rocket") as causal validity. This forced an incorrect Type A (Over-selection) error, permanently zeroing the instance reward.

6 Conclusion

We presented a reinforcement learning framework using GRPO to enhance abductive reasoning in large language models for SemEval Task 12. Guided by a composite reward, our RL-tuned models demonstrated strong reasoning capabilities, outperforming zero-shot baselines. However, extending this with a DeBERTa-based verifier revealed a severe generalization gap. While seemingly beneficial on the validation set, the verifier acted as an indiscriminate additive mechanism during testing, systematically appending false positives and degrading perfect predictions into over-selection errors.

Crucially, our error analysis exposed a fundamental vulnerability in the official evaluation metric. A naive fallback heuristic that blindly assigned a default option to the base model's unresolved instances artificially inflated the final score. This exploit demonstrates that the current paradigm creates a flawed asymmetry, penalizing honest abstention more heavily than reckless guessing. Recognizing and rewarding safe abstention is not merely a theoretical preference; it is a critical requirement for deploying abductive reasoning systems in real-world scenarios where false positive judgments carry high risks. Future work may explore developing robust, pruning-capable verifiers (e.g., via dynamic hard-negative oversampling) and advocate for evaluation metrics where the expected payoff of brute-force guessing remains strictly lower than that of principled reasoning.

Acknowledgements

We would like to thank Dr. Çağrı Çöltekin for his valuable guidance. We also gratefully acknowledge the University of Tübingen and bwUniCluster for providing the computing resources, and the SemEval-2026 Task 12 organizers for holding this shared task.

References

Pengfei Cao, Mingxuan Yang, Yubo Chen, Chenlong Zhang, Mingxuan Liu, Kang Liu, and Jun Zhao. 2026. Semeval-2026 task 12: Abductive event reasoning: Towards real-world event causal inference for large language models. *Preprint*, arXiv:2603.21720.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Daniel Han, Michael Han, and Unsloth Team. 2023. Unsloth.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y. Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. 2023. Language models can improve event prediction by few-shot abductive reasoning. *Preprint*, arXiv:2305.16646.

Nengbo Wang, Xiaotian Han, Jagdip Singh, Jing Ma, and Vipin Chaudhary. 2025. Causalrag: Integrating causal graphs into retrieval-augmented generation. *Preprint*, arXiv:2503.19878.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

A System Prompts and Input Templates

A.1 Prompt Template for Generative Models

The following structure illustrates the full prompt template used for both the GRPO training phase and the general inference phase across our generative models. To ensure logical consistency, the thinking process is force-started with the `<think>` token. Note: We show the prompts wrapped in Qwen's chat template, for the Ministral-3-8B and DeepSeek-R1-Distill-Llama-70B models, identical prompts were wrapped using their respective native chat templates.

```
<|im_start|>system
You are an expert in Abductive Event
    ↪ Reasoning. Your task is to
    ↪ analyze the provided context
    ↪ documents and identify **all and
    ↪ only** the direct causes of the
    ↪ target event among the given
    ↪ options (A, B, C, D).

Important rules:
- There may be one or more correct
    ↪ answers.
- You MUST NOT select any incorrect
    ↪ option - even one wrong choice
    ↪ results in zero reward.
- You MUST select every correct option
    ↪ - omitting a correct one reduces
    ↪ reward unless you avoid all
    ↪ errors.
- Base your judgment strictly on
    ↪ evidence in the context
    ↪ documents. Do not use external
    ↪ knowledge.
```

```
Output ONLY a valid JSON object with
    ↪ keys A, B, C, D. Each value must
    ↪ be: {"reasoning": "concise
    ↪ justification", "is_correct":
    ↪ true/false}.

[PHASE_SPECIFIC_LENGTH_CONSTRAINTS]

DO NOT include any text outside the
    ↪ JSON. DO NOT use
    ↪ markdown.<|im_end|>
<|im_start|>user
Context Documents:
{context_text}

Target Event: {target_event}
Options:
{options_text}

Please provide your reasoning and final
    ↪ judgment in the specified JSON
    ↪ format.<|im_end|>
<|im_start|>assistant
<think>
```

Training Directives During the GRPO training phase, the `[PHASE_SPECIFIC_LENGTH_CONSTRAINTS]` placeholder is populated with the following standard structural boundaries:

```
!!!IMPORTANT: Be concise, Keep your
    internal thinking under 2048 tokens.
!!!IMPORTANT: Completions longer than
    8192 tokens will be truncated!
!!!IMPORTANT: Follow the token
    restriction above or you will be
    penalized!!!
```

Inference Directives During the standalone inference phase, we enforce stricter length-constraint directives. Qualitative observations indicate that these constraints are necessary to mitigate the reasoning models' tendency to over-generate intermediate steps, thereby preventing runaway chain-of-thought loops and fatal context truncations:

```
!!!IMPORTANT: Keep your internal
    thinking and reasoning under 2048
    tokens. The entire response must fit
     within 3072 tokens!!!
!!!Be concise. Long-winded thinking and
    reasoning will be penalized!!!
```

A.2 LLM Soft Filtering (LLMSF) Prompt

To compress the raw retrieved documents prior to GRPO training (as described in Section 3.2), we utilized Qwen3-30B-Instruct-2507 with the following prompt structure to retain only causally relevant context. The prompt explicitly defines inclusion and exclusion criteria to standardize the reasoning filter:

```
<|im_start|>system
You are an expert data filter for an
    ↪ Abductive Event Reasoning task.
    ↪ Your job is to read a list of
    ↪ retrieved documents and select
    ↪ ONLY those that are relevant to
    ↪ the target events.
Criteria for KEEPING a document:
1. Direct Cause: Directly explains why
    ↪ the event happened.
2. Indirect Cause: Competitor actions,
    ↪ market trends, or regulatory
    ↪ changes that could trigger the
    ↪ event (Keep these!).
3. Context: Necessary background info
    ↪ to understand the entities
    ↪ involved.
Criteria for DISCARDING:
1. Totally irrelevant topics (e.g., a
    ↪ cooking recipe for a tech event).
2. Duplicate information that provides
    ↪ no new context.

Return the output as a JSON list of
    ↪ strings containing the IDs of the
    ↪ documents to keep. Example:
    ↪ ["doc-001", "doc-003"]<|im_end|>
<|im_start|>user
Topic Category: {topic_info}
Target Events:
{events_text}

Here are the Retrieved Documents:
{docs_text}

Which documents should be KEPT? Return
    ↪ JSON list only.<|im_end|>
```

A.3 DeBERTa Verification Template

For the verification stage, the text fields were concatenated using the `[SEP]` token to form a unified input for the cross-encoder:

```
Event: {target_event} [SEP]
Option: {opt_text} [SEP]
Qwen Prediction: {verdict_qwen} | Logic:
     {reasoning_qwen} [SEP]
Ref_Titles: {doc_titles}
```

B Detailed Hyperparameters

To facilitate full reproducibility, we report the granular hyperparameter settings for both the GRPO-optimized generative models (Table 6) and the DeBERTa-based verifier (Table 7).

GRPO Training (Qwen / Ministral)	Value
Hardware Environment	2 × NVIDIA A100 (80GB)
Precision	`bfloat16`
Max Sequence Length	16,384
Max Completion Length	8,192
Evaluated Checkpoint Step	492
Learning Rate	5e-7
Per Device Batch Size	4
Gradient Accumulation Steps	2
LoRA Rank (r) / Alpha (α)	64 / 64
Group Size (G)	4
Training Temperature (Sampling)	Model Default
Score Reward Base	+2.0 / +1.0 / 0.0
Format Reward	+0.5 / 0.0
Length Reward Bounds	[-0.8, +0.8]
Inference Temperature	0.0

Table 6: Hyperparameters for GRPO Training and Inference.

DeBERTa-v3-base Verification	Value
Hardware Environment	1 × NVIDIA GTX 1080Ti
Precision	`fp16`
Max Sequence Length	1,024
Learning Rate	2e-5
Maximum Training Epochs	6
Optimal Epoch (Early Stopping)	1.32
Per Device Batch Size	2
Gradient Accumulation Steps	16
Weight Decay	0.01
Hidden Dropout Probability	0.2
Attention Probs Dropout	0.2
Warmup Ratio	0.1

Table 7: Hyperparameters for DeBERTa Verifier Training.