HTMR: Hybrid Token Masking Reinforcement Learning with Verifiable Rewards for Event Argument Extraction with Multi-Perspective Reasoning

Jianwen Luo* , **Yongkang Jin*** , **Yu Hong†** , **Jianmin Yao**
School of Computer Science and Technology, Soochow University, Suzhou, China
{jwluo.ai, im.jinyongkang, tianxianer}@gmail.com

Abstract

Event Argument Extraction (EAE) aims to identify event arguments and assign semantic roles under a predefined schema. Recent work formulates EAE with large language models as a structured conditional generation task and applies Reinforcement Learning with Verifiable Rewards (RLVR) to optimize sequence-level event structures. However, RLVR-based EAE supervision is coarse-grained, as a single reward is assigned to the whole event structure, while optimization happens at the token level. This misalignment causes the same reward to be applied to all tokens, including those not related to event roles or arguments, introducing noise into the gradient updates and weakening the signals for decisions critical to argument extraction. To mitigate this misalignment, we propose Hybrid Token Masking RLVR (HTMR), which selectively updates policy gradients on both high-entropy forking tokens and event-critical tokens that define event structure, along with multi-perspective reasoning. Experiments across multiple benchmarks and models show that HTMR consistently outperforms full-token and high-entropy only RLVR methods. Moreover, HTMR transfers effectively as a plug-and-play approach to other tasks such as named entity recognition and relation classification. The code is publicly available for reproducibility.[1]

1 Introduction

Event Argument Extraction (EAE) aims to extract event arguments and assign corresponding semantic roles (Yang et al., 2024c). Figure 1 illustrates an example[2] from the ACE-2005 corpus (Doddington et al., 2004), where the model is required to extract entities involved in a Movement-Transport event and assign them appropriate semantic roles, such as classifying *agencies* as the **Agent**, *tourist* as the **Artifact** and *Middle East* as the **Destination**.

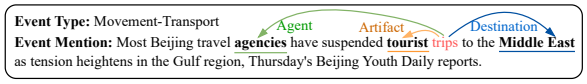

Figure 1: An illustrative example of EAE from ACE-2005 corpus, where *trips* serves as the event trigger.

Recent studies formulate EAE with Large Language Models (LLMs) as a structured conditional generation task, where LLMs directly produce argument spans under a predefined schema (Hong and Liu, 2024; Guo et al., 2024; Wang and Huang, 2024; Yang et al., 2024c; Wei et al., 2025). This formulation enables Reinforcement Learning with Verifiable Rewards (RLVR), which supports explicit reasoning prompts and deterministic output parsing, allowing LLMs to explore reasoning trajectories and achieve consistent performance gains through structure-level rewards (Luo et al., 2025).

However, existing RLVR-based EAE pipelines often rely on a single reasoning style during supervised warm-up, which can limit exploration diversity and hinder generalization. Following prior work on multi-perspective reasoning (Li et al., 2025), we incorporate multi-perspective reasoning into the warm-up stage. This initialization avoids over-constraining the LLMs with a fixed reasoning pattern, providing a better starting point for RLVR.

Despite these advances, existing RLVR-based approaches for EAE suffer from a fundamental misalignment between sequence-level supervision and token-level optimization. As illustrated in Figure 2(a), verifiable rewards in EAE are defined at the *sequence level*: a verifier evaluates the correctness of the entire predicted event structure and assigns a single scalar reward. However, policy-gradient optimization operates at the *token level*, updating the generation probabilities of individual tokens in the reasoning trace and structured out-

*These authors contributed equally to this work.
†Corresponding author.
[1]The source code is available at https://github.com/York-Gold/HTMR-EAE
[2]Event mention ID: XIN_ENG_20030314.0208-5-EV0.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19853–19873

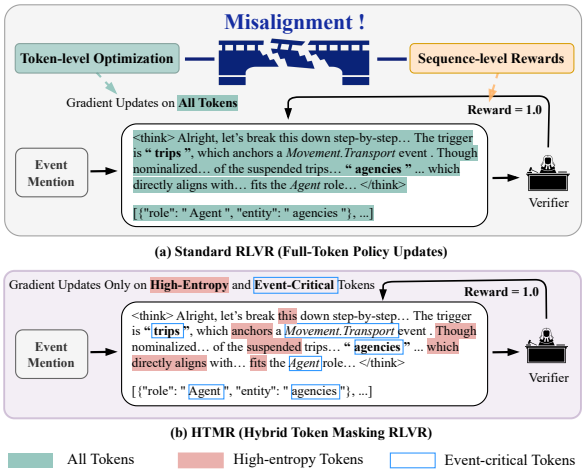

Figure 2: Misalignment between sequence-level rewards and token-level optimization in RLVR-based EAE, and how HTMR mitigates this misalignment. The complete model output is provided in Appendix A.

put. This misalignment leads to a broadcast-style credit assignment (Clark et al., 2021). Specifically, the same sequence-level reward is uniformly propagated to all generated tokens during training, regardless of whether they contribute to defining the event structure. Tokens related to linguistic fluency or intermediate reasoning receive learning signals comparable to those defining event types, role labels and argument spans, introducing noise into gradient updates for event-critical decisions.

One line of recent RLVR work addresses this issue by selectively applying policy updates to high-entropy tokens, which correspond to uncertain decision points during generation (Wang et al., 2025). This strategy concentrates learning signals on a small subset of "forking" tokens that dominate exploration in general reasoning tasks. However, as illustrated by the red-highlighted tokens in Figure 2(b), entropy-based token selection does not align well with EAE. Event-defining tokens typically exhibit relatively low entropy during generation.

To mitigate the fundamental misalignment, we propose Hybrid Token Masking RLVR (HTMR), which applies policy-gradient updates to both high-entropy "forking" tokens and event-critical tokens highlighted by the blue boxes in Figure 2(b). By grounding learning signals in event-defining decisions while retaining uncertainty-driven exploration, HTMR better bridges sequence-level supervision and token-level optimization. Experiments across multiple benchmarks and LLMs show that HTMR outperforms full-token and high-entropy only RLVR methods on EAE, delivering improved performance and stable optimization. Moreover, HTMR generalizes well to tasks such as named entity recognition and relation classification.

Our contributions can be summarized as follows:

- We identify a fundamental mismatch in RLVR-based EAE, where sequence-level structural rewards are optimized through token-level updates that are weakly aligned with event-defining decisions, leading to suboptimal policy updates.
- We propose HTMR, a task-aware post-training framework that integrates multi-perspective reasoning warm-up with hybrid token masking over high-entropy and event-critical tokens, explicitly anchoring policy updates to event-defining decisions and stabilizing RLVR optimization.
- We evaluate HTMR across multiple benchmarks and LLMs, showing consistent improvements over decoder-only LLM baselines, and conduct ablation and cross-task experiments to verify its effectiveness and generality beyond EAE.

2 Preliminaries

2.1 Token Entropy Calculation

Given an input x, an autoregressive LLM π_θ generates an output sequence $y = (y_1, \ldots, y_T)$. At decoding step t, conditioned on the prefix $y_{<t}$, the model produces a distribution over the vocabulary:

$$\boldsymbol{p}_t = \pi_\theta(\cdot \mid x, y_{<t}) = \mathrm{Softmax}\left(\frac{\boldsymbol{z}_t}{T}\right), \quad (1)$$

where $\boldsymbol{z}_t$ denotes the pre-softmax logits and T is the decoding temperature.

We quantify the model's uncertainty at position t using the token-level generation entropy:

$$H_t := -\sum_{j=1}^{|\mathcal{V}|} p_{t,j} \log p_{t,j}, \quad (2)$$

where $p_{t,j}$ is the probability of the j-th token in the vocabulary under the generation distribution $\boldsymbol{p}_t$.

A higher H_t indicates greater uncertainty over competing next-token candidates, while a lower H_t reflects confident predictions. Consistent with Forking Token Masking RLVR (FTMR)Wang et al. (2025), token entropy is defined as the entropy of the generation distribution at a position, rather than the sampled token itself.

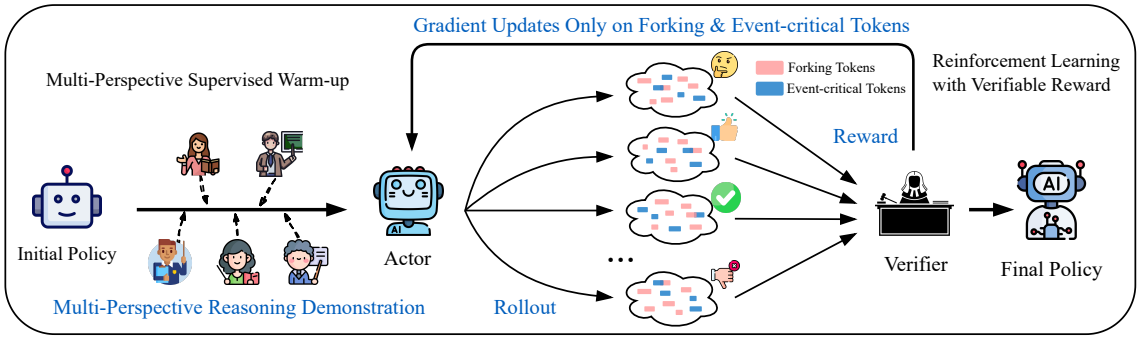

Figure 3: Overview of HTMR. The model is first warmed up on multi-perspective reasoning demonstrations, yielding a warmed-up actor with diverse, schema-consistent event reasoning paths. It is then optimized with RLVR, where policy-gradient updates are selectively applied to the union of high-entropy forking tokens and event-critical tokens. A verifier provides deterministic rewards based on structured EAE correctness, resulting in the final policy.

2.2 RLVR with DAPO

We adopt RLVR as the reinforcement stage of post-training, where rewards are automatically computed by checking the correctness of structured outputs. As the baseline optimizer, we employ Dynamic sAmpling Policy Optimization (DAPO) (Yu et al., 2025), a value-free RLVR algorithm with a token-level objective. DAPO performs policy optimization using the following clipped objective:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{\sum_{i=1}^{G}|\boldsymbol{o}^i|}\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{o}^i|}\min\Big(r_t^i(\theta)\,\hat{A}_t^i,\ \text{clip}\big(r_t^i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\,\hat{A}_t^i\Big)\Bigg], \tag{3}$$

where optimization is performed over groups containing both correct and incorrect responses. The definitions of $r_t^i(\theta)$, $\hat{A}_t^i$, and other internal parameters, along with descriptions of other RLVR algorithms and the reason for choosing DAPO as our baseline RL optimizer, are provided in Appendix B.

3 Approach

In this section, we first formalize EAE and its structured generation setup for reproducibility (§3.1). We next introduce multi-perspective reasoning warm-up that initializes the LLM with diverse, schema-consistent reasoning trajectories (§3.2). Subsequently, we present hybrid token masking for RLVR, updating policy gradients only on high-entropy forking and event-critical tokens (§3.3). Figure 3 illustrates the overall training pipeline.

3.1 Task Definition

Consistent with REAR (Luo et al., 2025), we formulate EAE as a structured conditional generation task. Given an input sentence describing a target event and optional auxiliary information (e.g., event type, trigger, entity mentions and role set), the model generates a structured representation of the event by assigning semantic roles to textual argument spans in the sentence. We adopt JSON schema for deterministic parsing. In addition to the standard setting, we also evaluate more challenging scenarios where entity mentions and trigger information are not provided in Section 4.4.3.

3.2 Multi-Perspective Reasoning Warm-Up

As depicted in Figure 3, HTMR adopts a two-stage post-training framework, applying SFT to initialize diverse, schema-consistent reasoning behaviors before RLVR. Concretely, for each training instance, we randomly select one reasoning perspective from a predefined set of five perspectives and generate a corresponding Chain-of-Thought (CoT) demonstration.[3] The CoT reasoning data are generated with the *Qwen3-Max* model via the Alibaba Cloud Model Studio platform.[4] Each perspective analyzes the same input from a distinct reasoning angle while producing an identical structured output:

- **Trigger-centric filtering**, which treats the event trigger as the semantic anchor and constrains argument plausibility accordingly;
- **Role-centric contrastive reasoning**, which explicitly compares candidate entities against role definitions and against each other;
- **Linguistic–heuristic analysis**, which emphasizes syntactic structure, discourse prominence and surface linguistic cues in context;

[3] For reproducibility, the random perspective selection process is controlled by a fixed random seed of 42.

[4] https://modelstudio.alibabacloud.com

- **Cognitive-style reasoning**, which simulates human-like attention shifts, hypothesis revision and narrative coherence checking;
- **Causal–temporal forensics**, which reconstructs local event timelines and causal relations to validate role assignments under event dynamics.

Let $\mathcal{D}_{\text{mp}} = \{(x^{(n)}, y_{\text{cot}}^{(n)})\}$ denote the resulting multi-perspective reasoning dataset. We perform SFT on $\mathcal{D}_{\text{mp}}$ to obtain a warmed-up base policy π_{θ_0}. The objective of this stage is to expose the model to diverse reasoning perspectives, rather than directly optimizing extraction accuracy, thereby providing a better-conditioned initialization for RLVR.

Detailed descriptions of CoT demonstrations for the five perspectives are provided in Appendix C.

3.3 Hybrid Token Masking RLVR

In the second stage of HTMR, the warmed-up actor (policy π_θ) is optimized via RLVR under a hybrid token masking scheme. As shown in Figure 3, the actor rollouts multiple responses, which are evaluated by a verifier to produce deterministic rewards. Policy-gradient updates are applied only to a selected subset of tokens.

Verifiable Reward. Consistent with REAR (Luo et al., 2025), we adopt sparse, terminal-state rewards for EAE. Let $\boldsymbol{y}$ and $\boldsymbol{y}'$ denote the predicted and gold sets of event arguments, respectively. The reward function is defined as:

$$\mathcal{R}_{\text{EAE}}(\boldsymbol{y}, \boldsymbol{y}') = \begin{cases} 1, & \text{if } |\boldsymbol{y}| = 0 \wedge |\boldsymbol{y}'| = 0, \\ \mathcal{F}(\boldsymbol{y}, \boldsymbol{y}'), & \text{if } |\boldsymbol{y}| > 0, \\ 0, & \text{on exception}, \end{cases} \tag{4}$$

where exceptions denote structurally invalid, non-parsable outputs, and $\mathcal{F}(\boldsymbol{y}, \boldsymbol{y}')$ is defined as:

$$\mathcal{F}(\boldsymbol{y}, \boldsymbol{y}') = \max\left(\frac{|\boldsymbol{y} \cap \boldsymbol{y}'|}{\max(|\boldsymbol{y}|, |\boldsymbol{y}'|)}, \sigma - \mu \cdot \max(|\boldsymbol{y}|, |\boldsymbol{y}'|), 0.1\right), \tag{5}$$

with $\sigma = 0.4$ and $\mu = 0.1$. We adopt this reward formulation following prior RLVR-based approach for EAE (Luo et al., 2025).

Hybrid Token Masking. For a batch $\mathcal{B}$, the actor samples a group of responses $\{\boldsymbol{o}^i\}_{i=1}^G$. For each token o_t^i, its generation entropy H_t^i is computed. Following Wang et al. (2025), the top-ρ high-entropy tokens are selected within each batch, corresponding to the red blocks in the model-generated responses shown in Figure 3. In our all main experiments, we set the factor $\rho = 20\%$, and the effect of different ρ values is analyzed in Section 4.4.2.

In parallel, event-critical tokens are identified from the generated response. Specifically, event roles, argument spans, event types and trigger mentions are extracted via deterministic parsing of the JSON-structured output and heuristic matching. Tokens corresponding to role labels, argument mentions, event types and trigger mentions are aligned to the response and marked as event-critical. The resulting event-critical tokens correspond to the blue blocks in Figure 3. We define an indicator function $\mathbb{I}_{\text{event}}(o_t^i)$ that equals 1 if token o_t^i belongs to any event-critical span, and 0 otherwise.

The hybrid token mask m_t^i is defined as the union of high-entropy and event-critical tokens:

$$m_t^i = \mathbb{I}\left[H_t^i \geq \tau_\rho^{\mathcal{B}}\right] \vee \mathbb{I}_{\text{event}}(o_t^i), \tag{6}$$

where $\tau_\rho^{\mathcal{B}}$ denotes the entropy threshold corresponding to the top-ρ fraction of tokens in batch $\mathcal{B}$.

Masked Policy Optimization. Based on the DAPO objective in Eq. 3, policy-gradient updates are restricted to tokens selected by the hybrid mask. For each input in a batch $\mathcal{B}$, the actor samples a group of G responses $\{\boldsymbol{o}^i\}_{i=1}^G$, and group-relative advantages $\hat{A}_t^i$ are computed from verifiable rewards following the DAPO formulation. The hybrid mask m_t^i determines whether token o_t^i contributes to the policy-gradient loss. The resulting batch-level optimization objective is given by:

$$\mathcal{J}_{\text{HTMR}}^{\mathcal{B}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^G \sum_{t=1}^{|\boldsymbol{o}^i|} m_t^i} \sum_{i=1}^G \sum_{t=1}^{|\boldsymbol{o}^i|} m_t^i \cdot \min\left(r_t^i(\theta)\hat{A}_t^i,\ \text{clip}(r_t^i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_t^i\right)\right], \tag{7}$$

where the normalization term in the denominator accounts only for tokens with $m_t^i = 1$, ensuring that gradient magnitudes are comparable across batches with different numbers of selected tokens. The definitions of $r_t^i(\theta)$, $\hat{A}_t^i$, and other internal parameters are provided in Appendix B.5. Algorithm 1 in Appendix B.5 summarizes the overall training procedure of HTMR.

4 Experimentation

In this section, we aim to explore the following Research Questions (RQs) regarding our HTMR:

- **RQ1:** Does HTMR consistently improve EAE performance over standard RLVR baselines across multiple benchmarks and LLMs? (§4.2)
- **RQ2:** How does HTMR compare with representative recent LLM-based EAE methods? (§4.3)

Approach	ACE05-E			ACE05-E$^+$			ERE			Average		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
Meta-Llama-3-8B-Instruct												
Original Model	27.7	38.7	32.3	28.3	34.4	31.0	39.1	43.5	41.2	31.7	38.9	34.8
Supervised Warm-up	67.4	61.3	64.2	62.0	65.5	63.7	74.9	72.4	73.6	68.1	66.4	67.2
DAPO (Yu et al., 2025)	67.3	68.3	67.8	66.5	67.8	67.1	75.3	75.3	75.3	69.7	70.5	70.1
FTMR (Wang et al., 2025)	69.0	69.6	69.3	68.5	68.7	68.6	76.1	77.4	76.8	71.2	71.9	71.6
HTMR (Ours)	**71.2**	**70.1**	**70.7**	**70.4**	**71.3**	**70.9**	**76.9**	**77.8**	**77.3**	**72.8**	**73.1**	**73.0**
DeepSeek-R1-Distill-Llama-8B												
Original Model	30.0	57.8	39.5	32.1	53.4	40.1	42.9	66.5	52.1	35.0	59.2	43.9
Supervised Warm-up	68.1	60.4	64.0	67.1	61.1	64.0	74.1	70.7	72.4	69.8	64.1	66.8
DAPO (Yu et al., 2025)	68.9	62.2	67.5	67.6	67.6	67.6	74.3	73.6	73.9	70.3	67.8	69.7
FTMR (Wang et al., 2025)	68.6	68.1	68.3	70.0	68.2	69.1	75.0	77.8	76.4	71.2	71.4	71.3
HTMR (Ours)	**70.8**	**70.3**	**70.5**	**71.0**	**69.5**	**70.2**	**76.9**	**80.8**	**78.8**	**72.9**	**73.5**	**73.2**
Qwen3-8B												
Original Model	42.0	**71.8**	53.0	43.4	65.5	52.2	60.1	77.0	67.5	48.5	71.4	57.6
Supervised Warm-up	65.4	62.8	64.1	65.7	59.4	62.3	71.1	71.1	71.1	67.4	64.4	65.8
DAPO (Yu et al., 2025)	68.3	69.6	69.0	68.1	70.1	69.1	75.3	71.5	73.4	70.6	70.4	70.5
FTMR (Wang et al., 2025)	**70.9**	70.0	**70.4**	69.8	71.6	70.7	**76.1**	74.5	75.3	**72.3**	72.0	72.1
HTMR (Ours)	70.0	70.8	**70.4**	**70.7**	**72.1**	**71.4**	75.1	**79.5**	**77.2**	71.9	**74.1**	**73.0**

Table 1: Main results on ACE05-E, ACE05-E$^+$ and ERE. Average denotes the mean P/R/F1 over the three datasets.

- **RQ3:** How do supervised warm-up and the number of CoT reasoning perspectives influence downstream EAE performance? (§4.4.1)
- **RQ4:** How do token selection strategies and the update top ratio (ρ) jointly influence the effectiveness of selective RLVR for EAE? (§4.4.2)
- **RQ5:** How does removing entity and event trigger information affect HTMR? (§4.4.3)
- **RQ6:** Can HTMR serve as a general plug-and-play RLVR strategy for NLP tasks beyond EAE with different prediction structures? (§4.5)

4.1 Experimental Setup

Benchmarks. We evaluate HTMR on three event understanding benchmarks: ACE05-E, ACE05-E$^+$ and ERE. ACE05-E and ACE05-E$^+$ are two standard variants derived from the ACE2005 corpus (Doddington et al., 2004), while ERE comes from the Rich ERE dataset (Song et al., 2015). Following prior work (Yang et al., 2024c; Luo et al., 2025), we adopt standard preprocessing to support sentence-level EAE evaluation. We report micro-averaged Precision (P), Recall (R) and F1 for argument–role assignments under exact span match, where both the argument span and its semantic role must be correctly classified.

Baselines. We compare HTMR with several representative baselines under a unified JSON-style structured generation schema: (i) Original Model, which performs prompt-only inference without post-training; (ii) Supervised Warm-up, which applies SFT on CoT demonstrations; (iii) DAPO (Yu et al., 2025), a full-token RLVR method; and (iv) Forking Token Masking RLVR (FTMR) (Wang et al., 2025), which restricts RLVR updates to the top-20% high-entropy forking tokens.

We report results of **HTMR (Ours)** alongside these baselines to evaluate the effectiveness of hybrid token masking that combines entropy-based and event-critical token selection.

LLMs. We experiment with multiple open-source backbones, including Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024), DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI, 2025)[5] and Qwen3-8B (Yang et al., 2025). All models follow the same decoding format and evaluation protocol.

More experimental setup details are provided in Appendix D and our open-source implementation.

4.2 Overall Experimental Results

In response to **RQ1**, Table 1 reports the results on ACE05-E, ACE05-E$^+$ and ERE across 3 backbone LLMs. Overall, HTMR consistently achieves the best or near-best performance across datasets and models, demonstrating its effectiveness for EAE.

For fair comparison, all RLVR-based methods (DAPO, FTMR and HTMR) use identical training and inference configurations, detailed configurations are provided in Appendix D.3.

Selective RLVR with FTMR outperforms full-token DAPO, indicating that reasoning-oriented learning is driven by a small set of high-entropy

[5] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

Method	Backbone Model	ACE05-E	ACE05-E$^+$	ERE
G-PTLM (Lin et al., 2023)	GPT-J-6B (Wang, 2021)	–	31.2	29.6
ChatGPT-IE (Han et al., 2023)	GPT-4 (Achiam et al., 2023)	34.4	36.3	–
Code4Struct (Wang et al., 2023)	Text-Davinci-003 (Brown et al., 2020)	60.4	–	–
RLQG (Hong and Liu, 2024)	LLaMA-2-13B (Touvron et al., 2023)	41.5	–	–
	Qwen-7B (Yang et al., 2024a)	43.2	–	–
Code4UIE (Guo et al., 2024)	Text-Davinci-002 (Brown et al., 2020)	57.0	–	–
Debate-EE (Wang and Huang, 2024)	Meta-Llama-3-8B-Instruct + GPT-3.5 (Ouyang et al., 2022)	56.0	–	–
	Gemini-Pro (Team et al., 2024) + GPT-3.5	59.5	–	–
Scented-EAE (Yang et al., 2024c)	Meta-Llama-3-8B-Instruct	62.4	61.9	60.6
CAT (Wei et al., 2025)	Qwen2.5-7B (Yang et al., 2024b)	55.6	53.3	46.4
REAR (Luo et al., 2025)	DeepSeek-R1-Distill-Llama-8B	63.3	64.1	62.2
HTMR (Ours)	Meta-Llama-3-8B-Instruct	**70.7**	70.9	77.3
	DeepSeek-R1-Distill-Llama-8B	70.5	70.2	**78.8**
	Qwen3-8B	70.4	**71.4**	77.2

Table 2: Comparison with recent baselines. All values are F1 scores (%). "–" indicates the dataset is not reported.

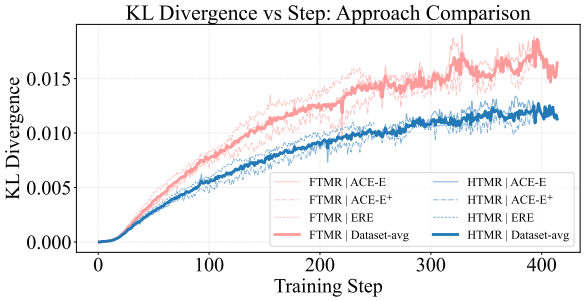

Figure 4: Policy KL divergence during training under different token masking strategies. Thin curves denote averages over three backbone models for each dataset, and the thick curve shows the overall average.

tokens. However, its effectiveness in EAE is limited, as entropy poorly aligns with event-defining tokens: critical elements have low entropy, while high-entropy tokens mainly capture intermediate reasoning rather than event-role decisions.

By explicitly incorporating event-critical tokens into selective policy updates, HTMR effectively mitigates the misalignment. As shown in Table 1, HTMR consistently outperforms FTMR across all datasets and backbone models, improving average F1 score by 1.4, 1.9 and 0.9 points on Meta-Llama-3-8B-Instruct, DeepSeek-R1-Distill-Llama-8B and Qwen3-8B, respectively. These improvements are typically reflected in balanced gains in both precision and recall, indicating that HTMR strengthens supervision on role labels and argument spans while still preserving uncertainty-driven exploration at key reasoning steps.

This advantage is also reflected in the optimization dynamics. Figure 4 shows the evolution of policy KL divergence (Kullback and Leibler, 1951) between the RLVR-updated policy and the reference policy, where lower values indicate more stable and consistent policy updates during training. FTMR exhibits larger and more volatile KL divergence, suggesting overly aggressive updates. In contrast, HTMR yields lower and smoother KL divergence trajectories across datasets and backbone models, indicating more controlled policy evolution. Overall, these results suggest that aligning selective RLVR updates with event-defining tokens is critical for effective optimization in EAE. Appendix E.1 presents the full KL-divergence trajectories for all dataset–model combinations, while Appendix E.2 isolates the impact of token selection by controlling the number of updated tokens.

4.3 Comparison with Recent Work

In answering **RQ2**, Table 2 compares HTMR with representative recent methods for EAE. As these approaches differ in backbone models, training paradigms and evaluation settings, the comparison should be viewed as a high-level reference rather than a strictly controlled benchmark.

Overall, HTMR achieves substantially stronger performance across ACE05-E, ACE05-E$^+$ and ERE than previously reported results. Compared with prompt-only or constraint-based approaches such as G-PTLM, ChatGPT-IE, Debate-EE and CAT, HTMR demonstrates the clear advantage of task-specific post-training. Relative to prior post-training methods, including supervised adaptation in Scented-EAE and RLVR-based optimization in REAR, HTMR further improves performance by explicitly aligning selective token-level updates with event-defining structure. These results demonstrate that task-aware selective RLVR is an effective and practical approach for improving structured EAE with LLMs, as it better aligns token-level optimization with event-defining structure. All meth-

Token Selection Strategy	ACE05-E			ACE05-E$^+$			ERE			Average		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
HTMR (ρ=20%) (Ours)	**70.8**	70.3	**70.5**	71.0	69.5	70.2	76.9	**80.8**	**78.8**	**72.9**	**73.5**	**73.2**
FTMR (ρ=20%)	68.6	68.1	68.3	70.0	68.2	69.1	75.0	77.8	76.4	71.2	71.4	71.3
HTMR (ρ=10%)	67.2	70.5	68.8	70.9	69.4	70.1	75.8	74.9	75.4	71.3	71.6	71.4
FTMR (ρ=10%)	65.9	**71.0**	68.4	**72.6**	67.2	69.8	72.0	71.1	71.5	70.2	69.8	69.9
HTMR (ρ=30%)	66.7	69.8	68.2	72.0	**70.2**	**71.1**	77.3	75.3	76.3	72.0	71.8	71.9
FTMR (ρ=30%)	66.5	67.7	67.1	67.9	69.1	68.5	73.2	77.8	75.5	69.2	71.5	70.4
Event-critical only	67.1	65.4	66.2	68.8	68.2	68.5	**79.9**	69.9	74.6	71.9	67.8	69.8
Event-critical ∩ High-entropy	64.0	68.3	66.1	69.5	66.8	68.1	75.1	70.7	72.8	69.5	68.6	69.0
High-entropy + Random	69.4	69.3	69.3	71.8	67.2	69.4	75.0	75.3	75.2	72.1	70.6	71.3
Event-critical + Random	68.6	68.3	68.4	69.4	67.1	68.2	77.0	74.1	75.5	71.7	69.8	70.7
Random-only	66.0	69.1	67.5	67.5	65.7	66.6	68.3	64.0	66.1	67.3	66.3	66.7

Table 3: Ablation on token selection strategies for RLVR in EAE on DeepSeek-R1-Distill-Llama-8B. ρ denotes the top ratio of selected high-entropy tokens, and HTMR selects the union of high-entropy and event-critical tokens. Random baselines are matched in update budget. **Best** results are in bold and second-best results are underlined.

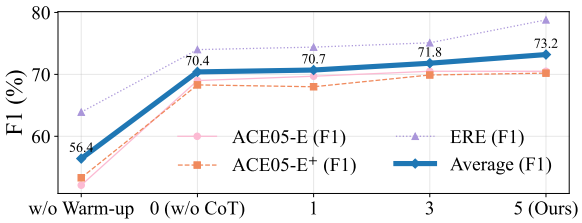

Figure 5: Effect of multi-perspective warm-up under different settings. "w/o Warm-up" denotes direct RLVR; x-axis numbers represent the number of perspectives.

ods listed in Table 2 are described in Section 5.

4.4 Ablation Studies

4.4.1 Effect of Multi-Perspective Warm-Up

Regarding **RQ3**, we analyze the effect of multi-perspective reasoning warm-up by varying the number of CoT perspectives. Figure 5 reports results on DeepSeek-R1-Distill-Llama-8B under different warm-up settings. Without supervised warm-up, performance is substantially degraded, indicating that RLVR alone is insufficient for structured EAE. Supervised warm-up without CoT already improves performance, while CoT reasoning yields further gains. Performance consistently increases with the number of perspectives across all datasets. Specifically, average F1 increases with more perspectives and peaks with five CoT perspectives, indicating that diverse yet schema-consistent warm-up provides a stronger initialization for RLVR.

4.4.2 Token Selection Strategies for RLVR

In addressing **RQ4**, we examine token selection strategies for RLVR in EAE. Table 3 compares entropy-based, structure-based and hybrid approaches under controlled update budget. Overall, HTMR consistently performs best, underscoring the importance of jointly updating high-entropy and event-critical tokens for RLVR in EAE.

We compare HTMR with FTMR under different values of the top-ρ ratio. HTMR consistently outperforms FTMR across all values of ρ, with the largest gain at ρ=20%. At smaller ratios (ρ=10%), FTMR overemphasizes a narrow set of high-entropy tokens, while HTMR preserves recall by explicitly updating event-defining tokens. As ρ increases to 30%, the performance gap narrows, indicating that updating more tokens partially mitigates entropy misalignment but reduces selectivity.

We further examine structure-driven and random baselines to disentangle the sources of improvement. Updating only event-critical tokens yields reasonable performance on ERE but degrades on ACE-style datasets, indicating that structural supervision alone is insufficient for complex reasoning, as it ignores high-entropy forking tokens encoding informative intermediate reasoning steps. Selecting the intersection of event-critical and high-entropy tokens performs worse, as many decisive event tokens exhibit low entropy once a coherent event hypothesis is formed. Budget-matched random variants consistently underperform HTMR and purely random selection performs worst, confirming that HTMR's gains arise from principled alignment between token-level updates and event structure rather than from updating more tokens.

4.4.3 Prompt Sensitivity: Entity and Trigger

To answer **RQ5**, we analyze the impact of prompt components by progressively removing entity men-

Prompt Setting	ACE05-E			ACE05-E$^+$			ERE		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
DeepSeek-R1-Distill-Llama-8B									
HTMR (full prompt)	70.8	70.3	70.5	71.0	69.5	70.2	76.9	80.8	78.8
w/o entity	65.2	65.5	65.4	66.3	66.0	66.2	71.1	71.1	71.1
w/o entity + trigger	64.5	64.7	64.6	65.3	61.2	63.2	64.5	66.1	65.3
Qwen3-8B									
HTMR (full prompt)	70.0	70.8	70.4	70.7	72.1	71.4	75.1	79.5	77.2
w/o entity	65.2	69.5	67.3	69.8	69.2	69.5	72.7	73.6	73.2
w/o entity + trigger	63.1	67.1	65.0	65.2	68.2	66.7	64.5	69.0	66.7

Table 4: Prompt ablation on removing entity and trigger.

tions and trigger information, where *w/o* denotes *without*. Following common LLM-based EAE settings, we treat the *w/o entity* prompt as the standard prompt setting. As shown in Table 4, removing entity mentions results in consistent F1 degradation across datasets and backbones, and further removing trigger information leads to more severe performance drops. This indicates that entity and trigger cues play a key role in identifying event-critical tokens and anchoring hybrid-masked policy updates. Notably, even under the standard *w/o entity* setting, HTMR still outperforms all representative prior EAE methods reported in Section 4.3.

4.5 Generalization to Other NLP Tasks

As for **RQ6**, we examine whether HTMR generalizes beyond EAE to other NLP tasks with different prediction structures. We consider Named Entity Recognition (NER) on WNUT16 (Strauss et al., 2016) and Relation Classification (RC) on SemEval (Hendrickx et al., 2010). NER is formulated as a span-level extraction task that requires identifying entity mentions and their types, while RC is treated as a sentence-level classification task that predicts a relation label between two given entities. We report **micro-averaged** P, R and F1 for NER, and **macro-averaged** P, R and F1 together with Accuracy (Acc) for RC. Detailed experimental setups, including dataset statistics, verifiable reward design and task-critical token identification methods, are provided in Appendix D.2.

Across both backbone models and tasks, HTMR consistently achieves the best or near-best performance, outperforming full-token RLVR and high-entropy only masking RLVR. Table 5 summarizes the experimental results. These gains indicate that hybrid token masking effectively aligns token-level policy updates with task-defining decisions, even when the task structure differs from EAE. Notably, HTMR improves both extraction-oriented and classification-oriented tasks, suggesting that it functions as a plug-and-play enhancement for RLVR across diverse NLP formulations.

Method	WNUT16 (NER)			SemEval (RC)			
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	Acc(%)
DeepSeek-R1-Distill-Llama-8B							
baseline	18.3	45.1	26.0	38.5	38.3	31.5	40.0
SFT	57.4	46.9	51.7	66.5	75.5	69.5	73.4
DAPO	56.4	53.1	54.7	75.5	78.7	75.9	77.9
FTMR	57.0	55.4	56.2	**75.9**	72.0	73.0	76.6
HTMR	**58.1**	**56.9**	**57.5**	74.1	**81.8**	**76.3**	**78.7**
Qwen3-8B							
baseline	29.6	56.6	38.9	52.8	60.1	51.9	60.7
SFT	59.9	49.8	54.4	70.5	77.3	71.9	74.5
DAPO	55.3	56.4	55.9	75.5	80.4	77.2	77.1
FTMR	57.2	56.5	56.8	**80.4**	81.4	**80.6**	78.9
HTMR	**59.0**	**57.2**	**58.1**	76.5	**83.9**	78.7	**79.5**

Table 5: Generalization of HTMR to other NLP tasks.

HTMR further yields more stable RL training than FTMR on both NER and RC, as evidenced by consistently lower policy KL divergence (Appendix E.3). This further supports the general applicability of hybrid token masking as a task-agnostic strategy for stabilizing and improving RLVR.

5 Related Work

We briefly review recent decoder-only LLM approaches to EAE, focusing on prompting strategies and post-training methods.

Early LLM-based EAE methods mainly improve extraction through prompting and constraint-based reasoning. G-PTLM (Lin et al., 2023) performs zero-shot EAE by scoring prompt-transformed passages. It further enforces global constraints across arguments and events to refine predictions. ChatGPT-IE (Han et al., 2023) empirically studies LLMs such as GPT-4 on information extraction. It highlights both their capabilities and limitations under prompt-only settings. Debate-EE (Wang and Huang, 2024) further refines event extraction via an iterative multi-agent debate process. It incorporates diverse retrieval and adaptive conformal prediction without parameter updates. Recently, CAT (Wei et al., 2025) analyzes preference traps in unsupervised EAE and proposes a two-stage *think–choose* framework that mitigates them via prompting and constrained selection, achieving zero-shot performance without parameter updates.

Beyond prompt-only extraction, prior work either reformulates EAE as structured generation or introduces post-training signals to im-

prove extraction behavior. Code-centric methods such as Code4Struct (Wang et al., 2023) and Code4UIE (Guo et al., 2024) formulate EAE as code generation with explicit schema constraints, optionally augmented with retrieval for improved output validity. RLQG (Hong and Liu, 2024) applies RL during post-training to optimize question generation quality in QA-based event extraction.

Post-training has become an effective paradigm for adapting LLMs more directly to EAE. Scented-EAE (Yang et al., 2024c) enhances supervised post-training by explicitly modeling entity-type semantics with stage-customized mechanisms. This strengthens role–entity alignment. REAR (Luo et al., 2025) further explores RLVR–based post-training. It first warms up the model with reasoning-enhanced supervision and relation-aware support, and optimizes reasoning trajectories for argument role disambiguation. FTMR (Wang et al., 2025) shows that RLVR is dominated by a small fraction of high-entropy forking tokens and proposes selectively updating policy gradients on these tokens.

Building on this insight, we propose task-aware selective post-training that aligns entropy-driven exploration with event schema-critical supervision.

6 Conclusion

In this paper, we proposed HTMR, a task-aware selective RLVR framework for EAE. HTMR addresses the mismatch between sequence-level verifiable rewards and token-level policy optimization by restricting policy updates to the union of high-entropy forking tokens and event-critical tokens, while using multi-perspective reasoning warm-up to provide a stronger initialization for RLVR. Experiments on ACE05-E, ACE05-E$^+$ and ERE across multiple LLM backbones show that HTMR consistently outperforms full-token and entropy-only RLVR baselines, yields more stable optimization, and transfers effectively to other structured prediction tasks such as NER and RC.

In the future, we will extend HTMR beyond sentence-level EAE to document-level settings. We will develop document-level selective RLVR with discourse-aware critical token identification and richer verifiable rewards that capture global event consistency across a document. More broadly, we will explore combining HTMR with retrieval or memory mechanisms and generalizing it to more complex event understanding problems, such as joint event extraction and event graph construction.

Limitations

In this work, we focus on sentence-level EAE under commonly used standard benchmark settings with predefined schemas and deterministic verifiers. While our experiments show that HTMR generalizes to other structured tasks such as NER and RC, we do not explore more complex scenarios involving document-level reasoning, overlapping events, or dynamically evolving schemas, which remain interesting directions for future study.

Compared with SFT, RLVR-based post-training requires higher computational cost, primarily due to the need for repeated on-policy rollouts during exploration. Although verifier-based reward computation itself is lightweight, generating multiple trajectories per input makes RLVR more resource-intensive than standard SFT, which may limit its use under strict computational budgets.

Ethical Considerations

This work investigates EAE using LLMs and publicly available benchmark datasets that have been widely used in prior natural language processing research. To the best of our knowledge, the datasets and pretrained models employed in this study do not pose additional ethical concerns beyond those commonly associated with large-scale language modeling. All experiments are conducted for research purposes and follow standard practices of academic integrity and responsible reporting. The proposed approach is intended to advance methodological understanding of structured EAE and is not designed for deployment in safety-critical or high-stakes real-world applications.

Acknowledgment

The research is supported by National Science Foundation of China (62376182).

References

OpenAI Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, et al. 2023. Gpt-4 technical report.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

David G. Clark, L. F. Abbott, and SueYeon Chung. 2021. Credit assignment through broadcasting a global error vector. In *Neural Information Processing Systems*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

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

Yucan Guo, Zixuan Li, Xiaolong Jin, Yantao Liu, Yutao Zeng, Wenxuan Liu, Xiang Li, Pan Yang, Long Bai, Jiafeng Guo, et al. 2024. Retrieval-augmented code generation for universal information extraction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 30–42. Springer.

Ridong Han, Chaohao Yang, Tao Peng, Prayag Tiwari, Xiang Wan, Lu Liu, and Benyou Wang. 2023. An empirical study on information extraction using large language models. *arXiv preprint arXiv:2305.14450*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Zijin Hong and Jian Liu. 2024. Towards better question generation in QA-based event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9025–9038, Bangkok, Thailand. Association for Computational Linguistics.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Zhongqiu Li, Shiquan Wang, Ruiyu Fang, Mengjiao Bao, Zhenhe Wu, Shuangyong Song, Yongxiang Li, and Zhongjiang He. 2025. Mr-uie: multi-perspective reasoning with reinforcement learning for universal information extraction. *Vicinagearth*, 2.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

Zizheng Lin, Hongming Zhang, and Yangqiu Song. 2023. Global constraints with prompting for zero-shot event argument classification. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2527–2538, Dubrovnik, Croatia. Association for Computational Linguistics.

Jianwen Luo, Yu Hong, Shuai Yang, and Jianmin Yao. 2025. REAR: Reinforced reasoning optimization for event argument extraction with relation-aware support. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7957–7972, Suzhou, China. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, et al. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ERE: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 89–98, Denver, Colorado. Association for Computational Linguistics.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ben Wang. 2021. Mesh-Transformer-JAX: Model-Parallel Implementation of Transformer Language Model with JAX. https://github.com/kingoflolz/mesh-transformer-jax.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Sijia Wang and Lifu Huang. 2024. Debate as optimization: Adaptive conformal prediction and diverse retrieval for event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16422–16435, Miami, Florida, USA. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023. Code4Struct: Code generation for few-shot event structure prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3640–3663, Toronto, Canada. Association for Computational Linguistics.

Yunhao Wei, Kai Shuang, Zhiyi Li, and Chenrui Mao. 2025. How do LLMs' preferences affect event argument extraction? CAT: Addressing preference traps in unsupervised EAE. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19529–19543, Vienna, Austria. Association for Computational Linguistics.

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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Yu Yang, Jinyu Guo, Kai Shuang, and Chenrui Mao. 2024c. Scented-EAE: Stage-customized entity type embedding for event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5222–5235, Bangkok, Thailand. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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

A Visualization of High-Entropy and Event-Critical Tokens

Figure 6 presents the complete model-generated output corresponding to the illustrative example in Figure 1. For each decoding step, we compute token-level generation entropy and select the top-ρ high-entropy tokens within the batch. In this visualization, we set $\rho = 20\%$, consistent with the main experiments. High-entropy forking tokens are highlighted with red blocks. Event-critical tokens are highlighted with blue boxes and include event types, triggers, role labels and argument spans.

The figure shows a clear but incomplete alignment between token-level uncertainty and event-critical structure. Several event-critical tokens exhibit relatively low entropy once the model establishes a coherent event hypothesis, indicating confident structural decisions. At the same time, many high-entropy tokens correspond to intermediate reasoning steps, discourse-level phrasing, or local lexical variation that does not directly affect event argument correctness. This divergence explains why entropy-based token selection alone may emphasize tokens that contribute little to structural accuracy, and it motivates explicitly incorporating event-critical tokens into token selection for RLVR.

B RLVR Objectives

This appendix summarizes the RLVR objectives referenced in the Section 2.2 and 3. To ensure consistency with the Section 3, we adopt the same notation as in the paper: x denotes the input sentence (with optional event-related annotations), and y' denotes the gold structured output. Given x, the rollout (old) policy $\pi_{\theta_{\text{old}}}$ samples an output sequence $\boldsymbol{o} = (o_1, \ldots, o_{|\boldsymbol{o}|})$. At decoding step t, the token-level importance ratio is defined as:

$$r_t^i(\theta) = \frac{\pi_\theta(o_{i,t} \mid x, \boldsymbol{o}_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x, \boldsymbol{o}_{i,<t})}. \tag{8}$$

For group-based RLVR methods, $\pi_{\theta_{\text{old}}}$ samples a group of G responses $\{\boldsymbol{o}_i\}_{i=1}^G$ for the same input x, which are evaluated by a deterministic verifier to produce scalar rewards $\{R_i\}_{i=1}^G$.

B.1 PPO (Reference Objective)

Proximal Policy Optimization (PPO) (Schulman et al., 2017) is a classical policy-gradient algorithm that stabilizes training by constraining policy updates within a trust region defined by clipping. PPO typically relies on a separate critic (value) network to estimate advantages.

Its clipped surrogate objective is:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}\Big[\min\Big(r_t(\theta)\,\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t\Big)\Big], \tag{9}$$

where $\hat{A}_t$ is the advantage estimated by the critic network and ϵ is the clipping hyperparameter.

Although PPO is widely used, its reliance on a separate critic model increases GPU VRAM usage and training complexity, which can be a practical limitation for RLVR with LLMs under constrained computational budgets.

B.2 GRPO Objective

Group Relative Policy Optimization (GRPO) (Zhihong Shao, 2024) removes the critic network by estimating advantages using relative rewards within a group of sampled responses. For a group of G responses with rewards $\{R_j\}_{j=1}^G$, the group-relative advantage $\hat{A}_t^i$ is computed as:

$$\hat{A}_t^i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}. \tag{10}$$

GRPO applies the PPO-style clipped objective at the group level and adds an explicit KL-divergence penalty to regularize policy updates:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|o^i|}\min\Big(r_t^i(\theta)\hat{A}_t^i,\ \text{clip}(r_t^i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^i\Big)\Bigg] - \beta \cdot \mathbb{E}[\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})], \tag{11}$$

where $r_t^i(\theta)$ denotes the token-level importance ratio defined in Eq. 8, and the factor β controls the KL regularization strength.

While GRPO avoids a critic network, the explicit KL penalty introduces additional sensitivity to hyperparameter tuning.

B.3 DAPO (Baseline RLVR Optimizer)

Dynamic sAmpling Policy Optimization (DAPO) algorithm (Yu et al., 2025) further improves GRPO and serves as the *baseline RL optimizer* in our experiments. DAPO incorporates several key enhancements: (i) **clip-higher** for asymmetric clipping, (ii) **dynamic sampling** to ensure mixed-quality groups, (iii) **token-level policy gradient loss**, and (iv) **overlong reward shaping**. These modifications substantially improve training stability and empirical performance in RLVR.

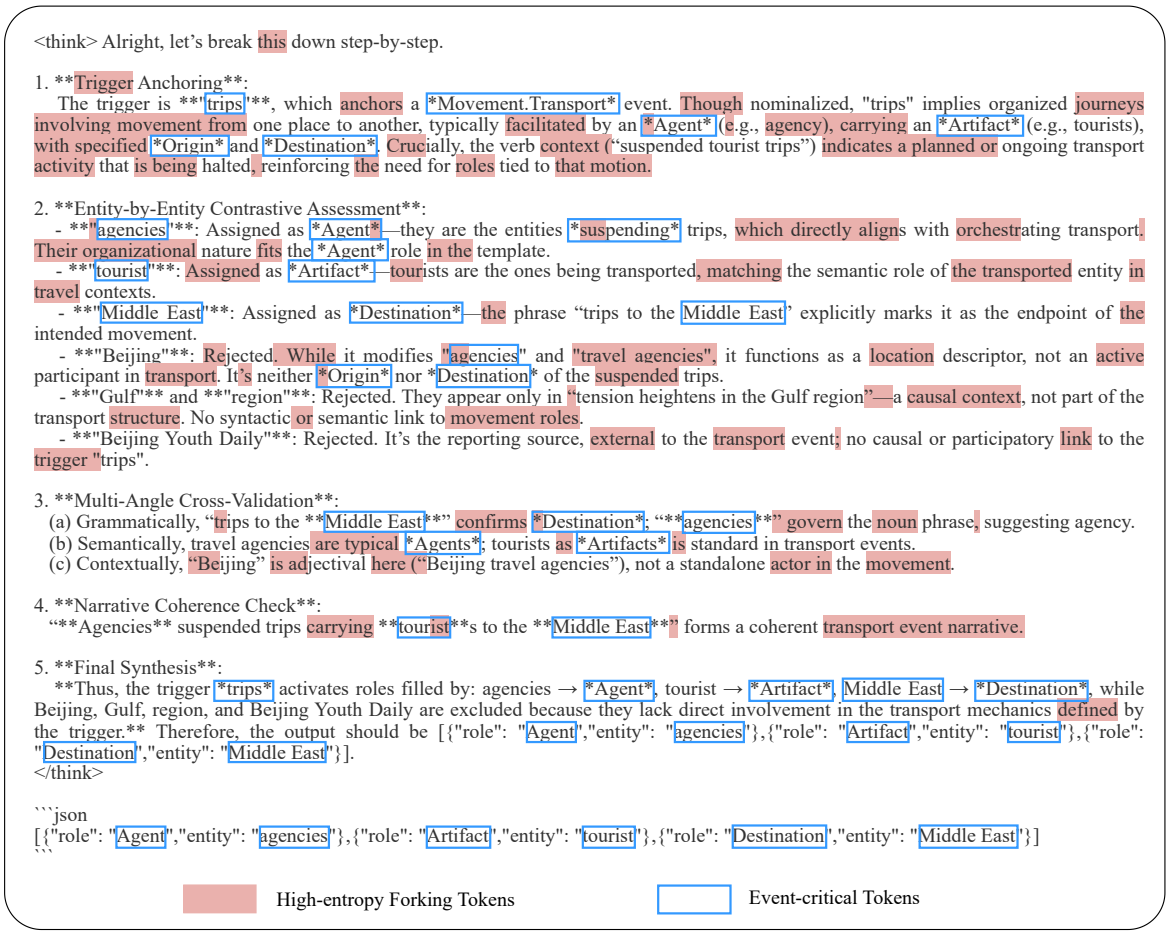

Figure 6: Complete model output for the example in Figure 1. Red blocks denote the top-20% high-entropy forking tokens, and blue boxes denote event-critical tokens such as event types, triggers, role labels and argument spans.

The DAPO objective used in our work is:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{\sum_{i=1}^{G}|\boldsymbol{o}^i|}\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{o}^i|}\min\Big(r_t^i(\theta)\,\hat{A}_t^i, \text{clip}\big(r_t^i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\,\hat{A}_t^i\Big)\Bigg], \tag{12}$$

where $r_t^i(\theta)$ denotes the token-level importance ratio defined in Eq. 8, and $\hat{A}_t^i$ is computed using the group-relative formulation in Eq. 10.

We choose DAPO as the baseline RL optimizer in our experiments for both practical and methodological reasons. Compared with PPO, DAPO does not rely on a separate critic network, which substantially reduces GPU memory consumption and simplifies the training pipeline. This design choice is particularly important for RLVR with large language models, where token-level policy optimization already incurs significant computational overhead. Compared with GRPO, DAPO incorporates several additional mechanisms—namely asymmetric clipping (clip-higher), dynamic sampling, token-level policy gradient normalization, and overlong reward shaping—that jointly improve optimization stability and mitigate degenerate update behaviors under sparse, verifiable rewards. As a result, DAPO provides a stronger and more reliable optimization baseline, allowing the effects of selective token masking in FTMR and HTMR to be evaluated under a stable and well-controlled RLVR framework.

B.4 FTMR (Forking Token Masking)

High-Entropy Forking Token Masking RLVR (FTMR) (Wang et al., 2025) restricts policy-gradient updates to high-entropy tokens that correspond to major branching decisions in the generation process. Let H_t^i denote the generation entropy of token o_t^i. For a batch $\mathcal{B}$, FTMR selects the top-ρ fraction of tokens by entropy, with threshold $\tau_\rho^{\mathcal{B}}$.

The FTMR objective is:

$$\mathcal{J}_{\text{FTMR}}^{\mathcal{B}}(\theta) = \mathbb{E}\Bigg[\frac{1}{\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{o}^i|}\mathbb{I}\big[H_t^i \geq \tau_\rho^{\mathcal{B}}\big]}\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{o}^i|}\mathbb{I}\big[H_t^i \geq \tau_\rho^{\mathcal{B}}\big]\cdot \min\Big(r_t^i(\theta)\hat{A}_t^i,\ \text{clip}\big(r_t^i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_t^i\Big)\Bigg]. \tag{13}$$

where H_t^i is the token-level generation entropy

defined in Eq. 2, $\mathbb{I}[\cdot]$ is the indicator function, $\rho \in (0, 1]$ controls the proportion of selected high-entropy tokens within batch $\mathcal{B}$, $\tau_\rho^{\mathcal{B}}$ is the corresponding entropy threshold, $r_t^i(\theta)$ is the token-level importance ratio defined in Eq. 8, and $\hat{A}_t^i$ denotes the group-relative advantage computed as in Eq. 10.

B.5 HTMR (Hybrid Token Masking, Ours)

HTMR extends FTMR by incorporating task-aware event-critical tokens. We define an indicator $\mathbb{I}_{\text{event}}(o_t^i)$ that equals 1 if o_t^i belongs to an event-critical span (event type, trigger, role label, or argument mention), and 0 otherwise.

The hybrid mask is defined as:

$$m_t^i = \mathbb{I}\big[H_t^i \geq \tau_\rho^{\mathcal{B}}\big] \vee \mathbb{I}_{\text{event}}(o_t^i). \tag{14}$$

Replacing the high-entropy only indicator in FTMR with m_t^i yields the HTMR objective:

$$\mathcal{J}_{\text{HTMR}}^{\mathcal{B}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_{i=1}^{G}\sum_{t=1}^{|o^i|} m_t^i} \sum_{i=1}^{G}\sum_{t=1}^{|o^i|} m_t^i \cdot \min\left(r_t^i(\theta)\hat{A}_t^i,\ \text{clip}(r_t^i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}})\hat{A}_t^i\right)\right] \tag{15}$$

where H_t^i denotes the token-level generation entropy defined in Eq. 2, $\tau_\rho^{\mathcal{B}}$ is the entropy threshold corresponding to the top-ρ fraction of tokens in batch $\mathcal{B}$, $\mathbb{I}_{\text{event}}(\cdot)$ indicates whether a token belongs to an event-critical span, m_t^i is the resulting hybrid token mask, $r_t^i(\theta)$ is the token-level importance ratio defined in Eq. 8, and $\hat{A}_t^i$ is the group-relative advantage computed as in Eq. 10.

Compared with FTMR, HTMR explicitly aligns selective token-level optimization with both uncertainty-driven reasoning branches and task-defining event structure, leading to more stable and effective RLVR for EAE.

Algorithm 1 summarizes the overall training procedure of HTMR, including multi-perspective supervised warm-up and hybrid-masked RLVR.

Algorithm 1 HTMR Training

Require: Multi-perspective dataset $\mathcal{D}_{\text{mp}}$, RLVR dataset $\mathcal{D}_{\text{rl}}$, top ratio ρ, group size G
Ensure: Final policy π_θ
Stage I: Supervised Warm-Up
Train base policy π_{θ_0} on $\mathcal{D}_{\text{mp}}$ via SFT
Initialize $\theta \leftarrow \theta_0$
Stage II: Hybrid Token Masking RLVR
for each batch $\mathcal{B}$ from $\mathcal{D}_{\text{rl}}$ **do**
  for each input $x \in \mathcal{B}$ **do**
    Rollout $\{\boldsymbol{o}^i\}_{i=1}^G \sim \pi_\theta(\cdot \mid x)$
    Compute task rewards $\{R^i\}_{i=1}^G$
    Compute token entropies $\{H_t^i\}$ and group-relative advantages $\{\hat{A}_t^i\}$
    Determine entropy threshold $\tau_\rho^{\mathcal{B}}$
    Mark event-critical tokens via schema-based parsing and span matching
    Construct hybrid token mask $m_t^i = \mathbb{I}[H_t^i \geq \tau_\rho^{\mathcal{B}}] \vee \mathbb{I}_{\text{event}}(o_t^i)$
    Update θ by optimizing the masked DAPO objective (HTMR objective, in Eq. 7)
  end for
end for
return π_θ

C Multi-Perspective Reasoning Prompts

Table 6 provides a detailed description of the five reasoning perspectives used in the multi-perspective supervised warm-up (Section 3.2). Each perspective enforces full entity coverage and schema-consistent outputs, while emphasizing a distinct inductive bias for event understanding. For reproducibility, the complete prompt templates are released in our open-source implementation.

D Experimental Setup Details

This appendix provides supplementary details on the benchmarks and experimental configurations used in our experiments, complementing the main descriptions in Section 4.1.

D.1 EAE Benchmarks

We conduct experiments on three benchmark datasets derived from the widely used ACE2005 corpus (Doddington et al., 2004) and the Rich ERE dataset (Song et al., 2015). The ACE2005 corpus contains 33 event types and 22 argument roles, while the ERE dataset defines 38 event types and 21 argument roles. Both datasets provide comprehensive annotations for events, entities and relations, making them well suited for sentence-level evaluation of EAE. Following established preprocessing procedures in prior work (Lin et al., 2020; Yang et al., 2024c; Luo et al., 2025), we construct two standard ACE-based variants, namely ACE05-E and ACE05-E$^+$ and adapt the ERE dataset to a unified structured generation format. Dataset statistics for these benchmarks are summarized in Table 7.

ACE05-E. ACE05-E is a standard benchmark for EAE with a closed set of event types and argument

Reasoning Perspective	Prompt-Derived Reasoning Description
Trigger-centric filtering *Core focus: Trigger semantics as the primary plausibility filter*	This perspective treats the event trigger as the semantic anchor that defines the core mechanics of the event. Reasoning begins by analyzing the trigger's action type, directionality, and affectedness to constrain which semantic roles can logically exist. Each candidate entity is then evaluated in a contrastive accept–reject manner against role requirements, with explicit justifications for both assignments and rejections. Role decisions are cross-validated through grammatical relations to the trigger, semantic compatibility with the event type, and contextual modifiers. A final narrative reconstruction step ensures that the selected roles form a coherent and plausible event description.
Role-centric contrastive reasoning *Core focus: Role semantics and inter-entity comparison*	This perspective anchors reasoning in the semantic definitions of event roles specified by the event template. For each entity, the model simulates its assignment to all plausible roles and explicitly compares its suitability against competing entities for the same role. The trigger and its surrounding context are used to disambiguate close candidates and eliminate semantically invalid pairings. Entities that do not fill any role are explicitly rejected with role-specific explanations, and the final configuration is synthesized to maximize global consistency with event semantics.
Linguistic–heuristic analysis *Core focus: Linguistic structure and discourse prominence*	This perspective adopts a forensic linguistic viewpoint, reconstructing event semantics from syntactic and discourse evidence. The trigger is analyzed as the semantic nucleus, with attention to tense, voice and dependency structure. Entity candidacy is assessed based on discourse salience, such as object position and prepositional attachment. Role assignments are filtered through template expectations and pragmatic event scripts grounded in world knowledge. All non-assigned entities are explicitly rejected, and a final coherence check accounts for negation, modality and reported speech.
Cognitive-style reasoning *Core focus: Human-like attention, hypothesis formation and revision*	This perspective simulates a human analyst's cognitive workflow when interpreting an event description. Reasoning starts with trigger-driven schema activation, followed by incremental role slot filling as entities are scanned in the sentence. Ambiguous entities give rise to competing hypotheses, which are evaluated and rejected using contextual cues such as prepositions, coreference and temporal markers. The sentence is then re-evaluated holistically to ensure narrative and causal coherence, yielding a final role assignment that reflects attention shifts and plausibility-driven revision.
Causal–temporal forensics *Core focus: Temporal ordering and causal consistency*	This perspective reconstructs the event by placing the trigger within a local temporal and causal timeline. Entities are interrogated using a three-lens framework that considers syntactic position relative to the trigger, semantic compatibility with role definitions and real-world plausibility. When multiple entities compete for a role, direct contrastive comparisons are performed to identify the most causally and contextually grounded assignment. The final role configuration is validated through narrative synthesis and strict adherence to the event template, avoiding unsupported or extraneous roles.

Table 6: Detailed description of the five reasoning perspectives used for multi-perspective supervised warm-up. Each perspective emphasizes a distinct inductive bias while enforcing schema-consistent structured outputs.

Dataset	Split	#Sents	#Entities	#Arguments
ACE05-E	Train	17,172	20,006	4,895
	Dev	923	2,451	605
	Test	832	3,017	576
ACE05-E$^+$	Train	19,216	47,554	6,607
	Dev	901	3,423	759
	Test	676	3,673	689
ERE	Train	8,886	22,831	4,372
	Dev	720	1,946	378
	Test	604	1,621	257

Table 7: Statistics of EAE datasets.

Dataset	Split	#Instances	#Entity Types	#Relations
WNUT16	Train	2,394	10	-
	Dev	1,000		
	Test	3,850		
SemEval	Train	6,507	-	19
	Dev	1,493		
	Test	2,717		

Table 8: Statistics of the NER and RC datasets.

roles. We follow the official train/dev/test splits and report micro-averaged precision, recall and F1 scores for argument-role assignments.

ACE05-E$^+$. ACE05-E$^+$ extends ACE05-E by introducing a broader and more fine-grained role inventory, which increases the difficulty of role disambiguation and schema generalization. The same evaluation protocol as ACE05-E is adopted.

ERE. ERE is an event extraction benchmark with a slightly different ontology and annotation style from ACE2005. We adapt our output schema to match the ERE role definitions and adopt the same evaluation protocol as ACE05-E and ACE05-E$^+$.

D.2 NER and RC Experimental Setup

In addition to EAE, we evaluate HTMR on other natural language processing tasks, specifically Named Entity Recognition (NER) and Relation Classification (RC), to assess its generalization ability. Specifically, we evaluate on two widely used benchmarks: WNUT16 (Strauss et al., 2016) for NER and SemEval-2010 Task 8 (SemEval) (Hendrickx et al., 2010) for RC. Both datasets provide sentence-level annotations and have been extensively used to study structured IE under different linguistic conditions. Dataset statistics for these benchmarks are summarized in Table 8.

D.2.1 Datasets

WNUT16. WNUT16 is a benchmark dataset for NER on user-generated text, with a predefined set of 10 entity types. We follow the official train/dev/test splits and report micro-averaged precision, recall and F1 scores under exact span match.

SemEval. SemEval is a widely used relation classification benchmark with 10,717 annotated samples, covering nine bidirectional semantic relation types and a special *no_relation* category and is evaluated using classification accuracy and macro-averaged precision, recall and F1.

D.2.2 Data Processing

We cast both NER and RC into a unified structured generation format to enable RLVR with verifiable rewards, consistent with our EAE setup.

NER formatting. For NER, we formulate the task as structured generation. Given an input sentence, the model is prompted to generate a JSON list of extracted entities, where each entity is represented by an `entity` field containing the entity mention text and a `type` field specifying the entity category, e.g., `["entity":"...", "type":"..."]`. Entity mentions are matched to the input sentence by exact string match, and outputs that are not valid JSON or do not conform to the predefined schema are treated as invalid generations and counted as incorrect predictions, since they cannot be parsed for structured evaluation.

RC formatting. For RC, we include the input sentence in the prompt, where the two target entity mentions are explicitly marked using special tags `<e1></e1>` and `<e2></e2>` to indicate the subject and object, respectively. The model is instructed to generate a JSON object containing a single relation type, represented by the `relation_type` field, e.g., `{"relation_type": "..."}`. The generated output is deterministically parsed for reward computation and evaluation, and outputs that are not valid JSON or do not conform to the predefined schema are treated as incorrect predictions.

Schema constraints. To ensure stable structured generation and verifier-based evaluation, we specify the label ontology and enforce a fixed output schema at the prompt level. This design is consistent with our EAE setup and facilitates deterministic parsing for reward computation and evaluation.

Additional prompt templates used in our experiments are provided in the released codebase.

D.2.3 Evaluation Protocol

NER metrics. For NER, we adopt the same evaluation metrics as in the EAE task. We report micro-averaged Precision (micro-P), Recall (micro-R) and F1 score (micro-F1), where a predicted entity is considered correct if and only if both the entity mention text and its entity type exactly match a gold annotation. Predictions are de-duplicated by `(entity, type)` pairs before scoring.

RC metrics. For RC, we report macro-averaged Precision (macro-P), Recall (macro-R) and F1 score (macro-F1) over all relation classes, and additionally report overall classification accuracy (Acc). A prediction is considered correct if the generated relation label exactly matches the gold label.

Decoding and reproducibility. For evaluation, we use deterministic (greedy) decoding with temperature 0. All reported results are computed on the official test sets using the same verifier-based parsing and matching rules as those applied during training, ensuring full consistency between reward computation and final evaluation.

D.2.4 Verifiable Rewards for NER and RC

We design task-specific verifiable reward functions for NER and RC, following the same verifier-based principle adopted for EAE (Section 3.3), in which rewards are computed via deterministic parsing of model outputs and exact comparison between predicted structured outputs and gold annotations.

NER reward. For NER, let s denote the model output string and $\mathcal{G}$ denote the gold entity set. We first check whether the output contains a valid <think>...</think> segment and whether the predicted entity list can be successfully parsed under the predefined JSON schema. If either condition fails, the reward is zero.

Let $\mathcal{P}$ be the parsed predicted entity set, where each entity is represented as a pair (e, t) of entity mention text and entity type.

The number of correctly predicted gold entities is computed using the following matching criterion:

$$C(\mathcal{P}, \mathcal{G}) = \sum_{(e,t)\in\mathcal{G}} \mathbb{I}\Big[\exists\,(e', t') \in \mathcal{P} :\ e = e' \wedge t = t'\Big]. \quad (16)$$

The final verifiable reward for NER is defined as the following piecewise function:

$$\mathcal{R}_{\text{NER}}(s, \mathcal{G}) = \begin{cases} 1, & \begin{array}{l} C(\mathcal{P},\mathcal{G}) = |\mathcal{G}| \\ \wedge\ |\mathcal{P}| = |\mathcal{G}|, \end{array} \\ \max\Big(\frac{C(\mathcal{P},\mathcal{G})}{|\mathcal{G}|},\, 0.1\Big), & \text{if } C(\mathcal{P},\mathcal{G}) > 0, \\ 0, & \text{on exception.} \end{cases} \quad (17)$$

This design assigns full credit only to exact set matches and provides a smoothed partial reward otherwise for partially correct predictions.

RC reward. For RC, let s denote the model output string and let r^* denote the gold relation type. As in NER, we first require that the output contains a valid <think>...</think> segment and that a JSON object with a `relation_type` field can be parsed from the output. If either of these format checks fails, the reward is set to zero. Let $r = \textsc{Parse}(s)$ denote the predicted relation type. The verifiable reward for RC is defined as:

$$\mathcal{R}_{\text{RC}}(s, r^*) = \begin{cases} 1, & r = r^*, \\ 0.1, & r \neq r^*, \\ 0, & \text{on exception.} \end{cases} \quad (18)$$

This sparse reward assigns full credit to exact relation classification and a small constant reward to incorrect but well-formed predictions, encouraging structured output while maintaining stable RL optimization throughout training.

D.2.5 Hybrid Token Masking

NER Hybrid Token Masking. Following the hybrid token masking strategy described in Section 3.3, we construct task-specific hybrid masks for NER by combining uncertainty-based and entity-critical signals. For a batch $\mathcal{B}$, the actor samples a group of G responses $\{\boldsymbol{o}^i\}_{i=1}^G$. For each generated token o_t^i, its token-level generation entropy H_t^i is computed, and tokens whose entropy ranks within the top-ρ fraction of the batch are selected as high-entropy tokens.

In parallel, entity-critical tokens are identified from the parsed structured output. For NER, entity-critical tokens correspond to both entity mention texts and entity type labels extracted from the predicted JSON entity list. These spans are deterministically aligned to the generated response and marked as entity-critical. We define an indicator function $\mathbb{I}_{\text{entity}}(o_t^i)$ that equals 1 if token o_t^i belongs to any entity-critical span, and 0 otherwise. The resulting hybrid token mask is defined as:

$$m_t^i = \mathbb{I}\big[H_t^i \geq \tau_\rho^{\mathcal{B}}\big] \ \vee\ \mathbb{I}_{\text{entity}}(o_t^i), \quad (19)$$

where $\tau_\rho^{\mathcal{B}}$ denotes the entropy threshold for the top-ρ fraction of tokens in batch $\mathcal{B}$, with ρ set to 20%. The hybrid mask m_t^i is then applied in the masked policy optimization objective in Eq. 7.

RC Hybrid Token Masking. For RC, we follow the same hybrid token masking strategy as in NER, while adapting the definition of critical tokens to relation classification. For a batch $\mathcal{B}$, the actor samples a group of G responses $\{\boldsymbol{o}^i\}_{i=1}^G$. For each generated token o_t^i, its token-level generation entropy H_t^i is computed, and tokens whose entropy ranks within the top-ρ fraction of the batch are selected as high-entropy tokens.

In parallel, relation-critical tokens are identified from the structured output and the given entity mentions. Specifically, relation-critical tokens include the generated relation type label (i.e., the value of the `relation_type` field) as well as the two target entity mention texts involved in the relation. These components are deterministically parsed and aligned to the generated response. We define an indicator function $\mathbb{I}_{\text{rel}}(o_t^i)$ that equals 1 if token o_t^i belongs to any relation-critical span, and 0 otherwise. The resulting hybrid token mask is defined as the following logical combination:

$$m_t^i = \mathbb{I}\big[H_t^i \geq \tau_\rho^{\mathcal{B}}\big] \ \vee\ \mathbb{I}_{\text{rel}}(o_t^i), \quad (20)$$

where $\tau_\rho^{\mathcal{B}}$ denotes the entropy threshold for the top-ρ fraction of tokens in batch $\mathcal{B}$, with ρ set to 20%. The hybrid mask m_t^i is then used in the masked policy optimization objective in Eq. 7.

D.3 Experimental Configurations

This section summarizes the key hyperparameters used across different training stages in our experiments, including multi-perspective supervised warm-up, reinforcement learning with verifiable rewards based on DAPO, and its selective variants FTMR and HTMR. Unless otherwise specified, we adopt the same optimization and decoding configurations for DAPO, FTMR and HTMR to ensure a fair comparison, with differences arising only from the token masking strategy.

For inference and evaluation, we adopt vLLM (Kwon et al., 2023) as the decoding backend to accelerate large-scale generation. vLLM is a fast and easy-to-use library for large language model inference and serving, designed to support high-throughput and memory-efficient decoding. By leveraging efficient memory management and PagedAttention, vLLM enables low-latency and scalable inference, substantially improving the efficiency of both evaluation and reinforcement learning rollouts. This design allows us to generate long responses and multiple samples per prompt without incurring prohibitive memory overhead.

Training for all experiments is conducted on 4 NVIDIA H20 GPUs, each with 141 GB of memory, while inference and evaluation for each model are performed on a single NVIDIA H20 GPU. Our experiments are conducted on this hardware setup. Table 9 provides the key training and evaluation hyperparameters used in our experiments, along with their values and definitions. The full configuration details can be found in our open-source code.

E Supplementary KL-Divergence Visualizations of Selective RLVR

This appendix presents a detailed KL-divergence analysis of training dynamics under different selective token masking strategies for RLVR. The visualizations complement the main results by providing a fine-grained view of policy stability across datasets, backbone models and task settings.

The analysis covers three aspects. First, we report complete policy KL-divergence trajectories corresponding to the summarized trends in the main paper, enabling a full inspection of temporal training behavior. Second, we include controlled comparisons that match the number of updated tokens across methods, isolating the effect of token selection from update budget. Third, we extend the KL-divergence analysis to tasks beyond event argument extraction, assessing the generality of the stability properties of hybrid token masking. All models, datasets and training configurations follow the setups described in Section 4.1 and Appendix D.

E.1 Complete KL-Divergence Dynamics

Figure 7 presents the complete policy KL-divergence trajectories that underlie the summarized results in Figure 4. The figure consists of nine subplots, corresponding to all combinations of three datasets and three backbone models described in Section 4.1. In the figure, *LLaMA3-8B* refers to Meta-Llama-3-8B-Instruct and *D-LLaMA3-8B* refers to DeepSeek-R1-Distill-Llama-8B. Each subplot reports the evolution of policy KL divergence over training steps for different token masking strategies, using identical optimization settings.

Across datasets and backbone models, entropy-only masking with high-entropy Forking Token Masking RLVR (FTMR) consistently exhibits larger KL divergence and stronger temporal fluctuations. These patterns indicate more aggressive and less stable policy updates. In contrast, Hybrid Token Masking RLVR (HTMR) produces smoother KL-divergence curves with reduced variance across training steps. This behavior reflects more controlled policy updates and suggests that prioritizing event-critical tokens helps constrain policy drift while preserving effective learning signals.

E.2 Controlling for the Number of Tokens

A potential explanation for the improved training stability of HTMR is that it may update more tokens than high-entropy only masking. To isolate the effect of token selection from the effect of token quantity, we conduct a controlled comparison under the same experimental setup. We evaluate three strategies on DeepSeek-R1-Distill-Llama-8B and Qwen3-8B: FTMR, FTMR augmented with randomly selected tokens and HTMR. For FTMR+Random, we sample random tokens to match the total number of tokens optimized by HTMR, ensuring identical token budgets.

Figure 8 reports the resulting policy KL-divergence trajectories. Augmenting FTMR with randomly selected tokens leads to a modest reduction in KL divergence in some training phases, indicating that increasing the number of updated tokens can partially smooth policy updates. However, this effect remains limited and inconsistent, and in certain settings FTMR+Random exhibits comparable or even higher KL divergence than FTMR. In con-

Parameter	Value	Definition or Explanation
		Multi-Perspective Supervised Warm-up
$micro\_batch\_size$	2	Number of training examples per GPU before gradient aggregation
$gradient\_accum$	2	Steps of gradients accumulated before parameter updates
$batch\_size$	16	Effective global batch size for supervised warm-up
lr	1.0e-5	Learning rate for supervised warm-up
max_length	8192	Maximum input sequence length
$epochs$	3	Number of supervised warm-up training epochs
		Reinforcement Learning with Verifiable Rewards
$resp\_num$	4	Number of responses sampled per prompt during RL rollouts
$prompt\_max\_len$	2048	Maximum prompt length during RL optimization
$response\_max\_len$	4096	Maximum response length during RL optimization
$data\_truncation$	left	Truncation side for overlength inputs
rl_lr	2.0e-6	Learning rate for policy optimization in RL
lr_warmup_steps	10	Number of warm-up steps for RL learning rate
$weight\_decay$	0.1	Weight decay applied during RL optimization
$clip\_ratio\_low$	0.2	Lower clipping threshold for DAPO-style updates
$clip\_ratio\_high$	0.28	Upper clipping threshold for DAPO-style updates
$clip\_ratio\_c$	10.0	Lower bound constant for Dual-clip ratio clipping
$use\_kl\_in\_reward$	False	Disable in-reward KL penalty
use_kl_loss	True	Enable KL divergence as an explicit loss term in actor optimization
kl_loss_coef	0.1	Coefficient of the KL loss term
$train\_batch\_size$	8	Prompt-level batch size for policy updates
gen_batch_size	32	Batch size for response generation during rollouts
$ppo\_mini\_batch\_size$	32	Mini-batch size for policy updates
$max\_num\_gen\_batches$	2	Maximum number of generation batches used for group construction / filtering
$use\_remove\_padding$	True	Enable padding removal for efficiency in model forward / log-prob computation
$use\_dynamic\_bsz$	True	Enable dynamic batch sizing for actor / rollout / reference log-prob computation
$temperature$	0.9	Sampling temperature during RL rollouts
top_p	0.95	Nucleus sampling threshold during RL rollouts
$grad\_clip$	1.0	Gradient clipping threshold during RL optimization
$epochs$	3	Number of RLVR training epochs
		Model Inference and Evaluation
do_sample	False	Deterministic decoding for evaluation
$temperature$	0.01	Near-deterministic decoding temperature for evaluation
top_p	1.0	Nucleus sampling threshold during evaluation
$seed$	42	Random seed for reproducible generation
n	1	Number of generated outputs per prompt during evaluation
$best\_of$	1	Number of candidates considered before selecting the final output
$repetition\_penalty$	1.0	Repetition penalty factor
max_tokens	8192	Maximum number of tokens generated per output during evaluation

Table 9: The hyperparameters used in our experiment.

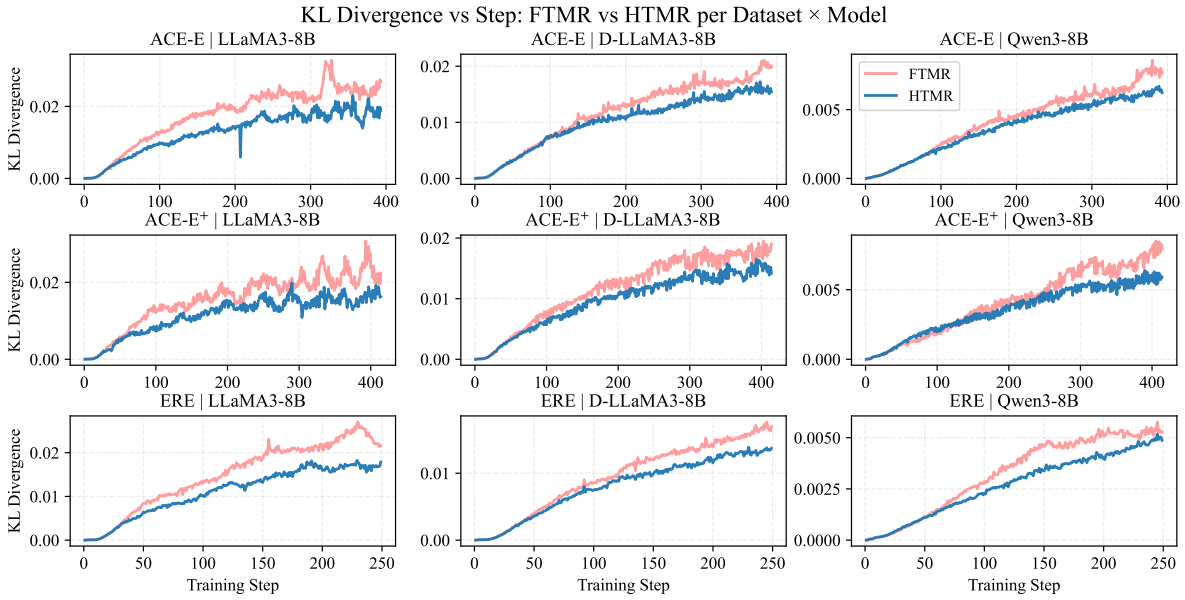

Figure 7: Complete policy KL-divergence training curves for all datasets and backbone models described in Section 4.1, providing the full experimental records underlying the summarized trends shown in Figure 4.

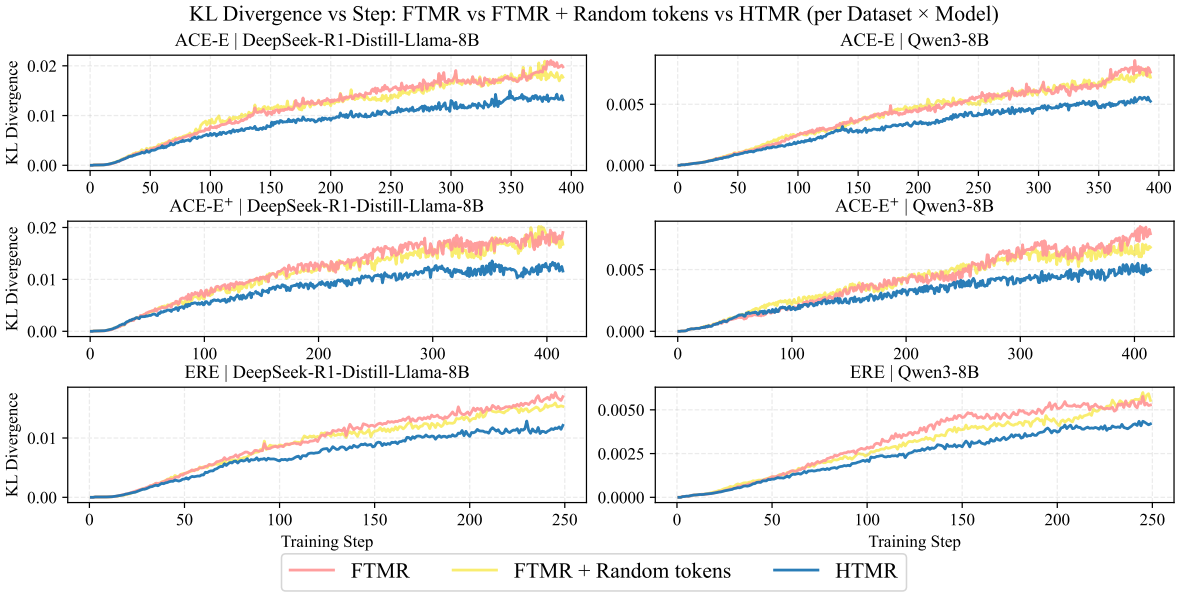

Figure 8: Policy KL-divergence trajectories for FTMR, FTMR augmented with randomly selected tokens (FTMR+Random Tokens), and HTMR on DeepSeek-R1-Distill-Llama-8B and Qwen3-8B. FTMR+Random Tokens matches HTMR in the number of optimized tokens, isolating the effect of token selection from token count.

trast, HTMR consistently produces the smoothest and lowest-variance KL-divergence curves across both backbone models. These results indicate that training stability does not primarily stem from updating more tokens, but from selectively prioritizing event-critical tokens during RLVR.

E.3 Cross-Task KL-Divergence Dynamics

Figure 9 extends our KL-divergence analysis beyond event argument extraction to additional Natural Language Processing (NLP) tasks, namely Named Entity Recognition (NER) on WNUT16 (Strauss et al., 2016) and Relation Classification (RC) on SemEval (Hendrickx et al., 2010). These tasks differ substantially from EAE in both output structure and supervision granularity, providing a complementary testbed for examining whether the stability advantages of hybrid token masking persist across task formulations.

The figure reports policy KL divergence over

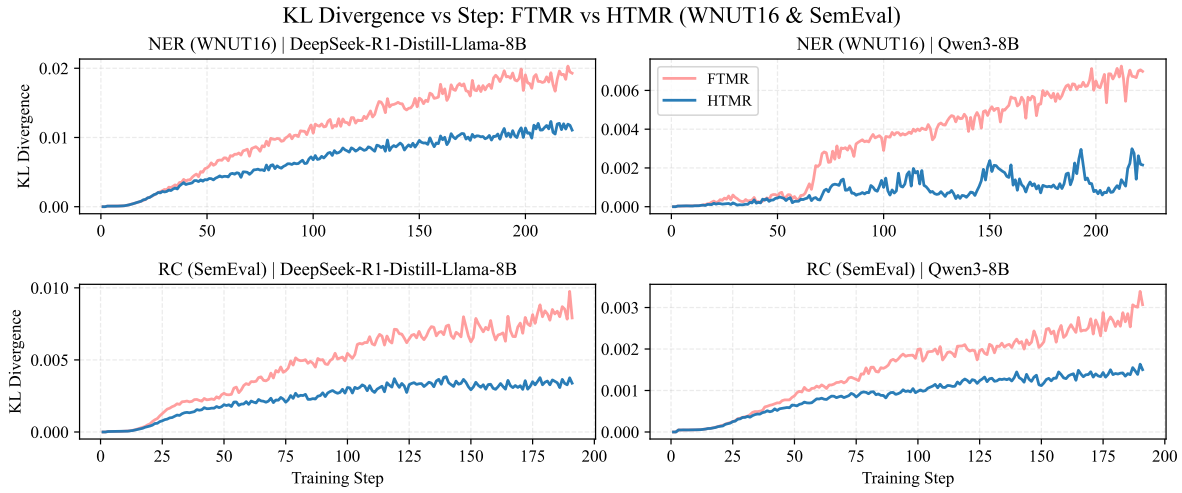

Figure 9: Policy KL-divergence trajectories of FTMR and HTMR on additional NLP tasks. The top row reports results on NER using WNUT16, and the bottom row reports results on RC using SemEval. HTMR consistently yields lower KL divergence than high-entropy only FTMR across tasks and backbone models.

training steps for two backbone models, DeepSeek-R1-Distill-Llama-8B and Qwen3-8B, comparing high-entropy only masking (FTMR) with hybrid token masking (HTMR). Across all task–model combinations, HTMR consistently yields lower KL divergence than FTMR, indicating more stable policy updates. This trend holds for both span-structured prediction in NER and label-structured prediction in RC, despite their different task formulations. Overall, these results suggest that anchoring selective policy updates to task-critical tokens leads to more controlled optimization under RLVR, and that this effect generalizes beyond EAE.