BALANCE-DEE: BALANCING EVENT CANDIDATE SPACE EXPANSION AND FALSE-POSITIVE CORRECTION FOR DOCUMENT-LEVEL EVENT EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Document-level event extraction (DEE) systems often miss non-salient events and make cross-event argument assignment errors. Expanding the event candidate space can recover missed events, but it also introduces spurious candidates and increases structural conflicts. We study this coverage–correction tradeoff and propose BALANCE-DEE, a structured refinement framework that jointly evaluates candidate-event validity and state-conditioned action gain. A discriminatively enhanced schema guides a frozen large language model to propose potentially missed events; a two-factor eventness assessment based on event instantiation and textual support filters unsupported candidates; and a learned action-gain function scores insertion, deletion, and split-and-reallocation edits relative to the current event-record state. The model iteratively executes positive-gain edits and recomputes state-conditioned gains after each update. Across FNDEE, CrudeOilNews, and SENTiVENT, BALANCE-DEE improves trigger and argument F_1 over the corresponding base extractor. On FNDEE, it raises overall event recall by 2.3 percentage points and non-salient event recall by 4.2 points, reduces over-extracted arguments by 13.2%, and lowers the cross-event argument attachment rate from 20.4% to 12.9%. These results show that high-recall candidate expansion is most effective when coupled with state-aware validation and structured correction.

1 INTRODUCTION

Document-level event extraction (DEE) aims to identify events from text spanning multiple sentences and extract their triggers and associated arguments. Because information about a single event is often dispersed across sentences, existing studies have improved cross-sentence information modeling and event-record construction through event-path expansion, heterogeneous-graph interaction, parallel prediction, and global set optimization (Zheng et al., 2019; Xu et al., 2021; Yang et al., 2021; Wang et al., 2023). However, these methods more readily identify salient events that are closely related to the document topic and supported by abundant semantic cues. They remain prone to missing non-salient events whose expressions are subtle, mentions are infrequent, or connections to other events are weak (Liu et al., 2018; Jindal et al., 2020).

As shown in Figure 1(a), the system-testing event is embedded in a subordinate expression describing the whole-aircraft flight-test event, with which it shares the subject, time, and event type. An existing extraction model (Xu et al., 2024) not only misses this non-salient event, but also incorrectly attaches its equipment argument to the whole-aircraft flight-test record. This example shows that a missed non-salient event can further induce argument extraction and assignment errors.

Expanding the event candidate space is a direct strategy for recovering missed events. Large language models (LLMs) offer broad semantic coverage and can identify weak triggers, subordinate expressions, and dispersed cues that conventional extractors tend to overlook. Open-ended generation, however, may also produce inconsistent predictions because of external knowledge interference and insufficient structural constraints, and may even generate content absent from the source text (Haji et al., 2025). Following the definition of spurious predictions in prior work (Bayer et al., 2021), we define an event produced by candidate expansion that cannot be matched to any gold event as a *spurious event*. Such events may arise from treating non-event information as an event, confusing

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

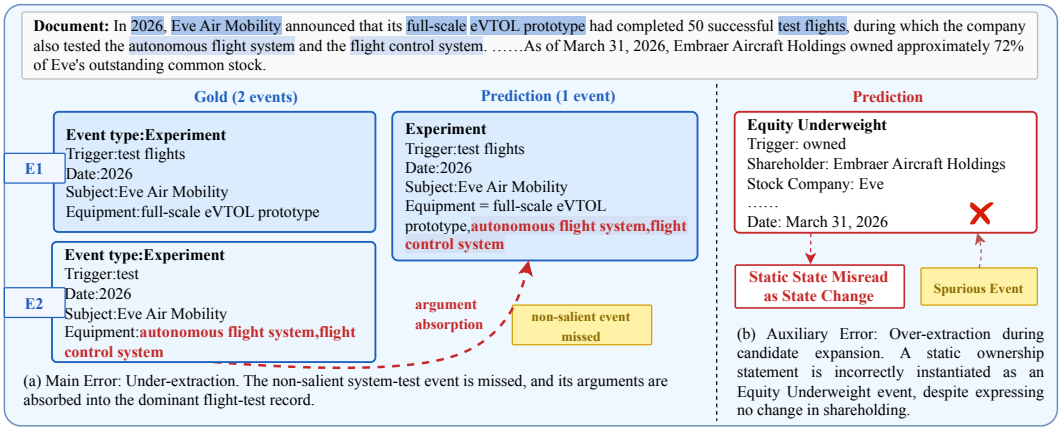

Figure 1: Coupled error modes in DEE.

event instances, or model hallucination. In Figure 1(b), for example, the source text only states that Embraer holds shares in Eve, whereas the model incorrectly extracts an EQUITY UNDERWEIGHT event.

These observations reveal an inherent tradeoff between event omissions caused by insufficient candidate coverage and spurious events introduced by candidate expansion. As illustrated in Figure 2, expanding beyond the region covered by the base extractor can recover missed non-salient events, but it increasingly overlaps with regions containing spurious events, creating an ambiguous boundary region. Candidate expansion therefore offers event-recovery gains while incurring false-positive risks and potentially exacerbating argument conflicts. Prior work has separately studied event omissions (Ji & Grishman, 2008; Hu et al., 2025), prediction correction and structural verification (Zhu et al., 2024; Guo et al., 2026), and multi-event relation modeling and cross-event argument confusion (Liu et al., 2024; Shuang et al., 2024). Most methods focus on a single error type and thus cannot jointly address event omissions, erroneous predictions, and argument misassignment.

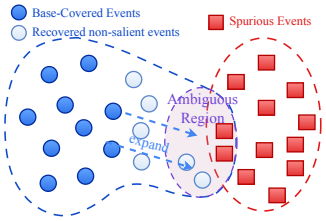

Figure 2: Recovery gains and spurious-event risks of candidate expansion.

To address this limitation, we propose a DEE method that balances event candidate space expansion against false-positive correction. The key idea is to learn an action-gain function for the ambiguous region created by candidate expansion, jointly accounting for the recovery benefit of an expanded candidate and the costs of its false-positive and argument-conflict risks. Relative to the current event records, the gain function dynamically evaluates the incremental contribution of each candidate-conditioned revision action to the overall extraction result. The model then iteratively updates the event records to improve event recall, suppress false positives, and mitigate cascading argument errors.

Our main contributions are as follows:

- To recover missed non-salient events, we propose a discriminatively enhanced schema-guidance mechanism that augments conventional schema descriptions with event-instance boundaries and criteria for distinguishing similar events, thereby improving candidate coverage of potentially missed events.
- To suppress false-positive events introduced by candidate expansion, we design a collaborative verification mechanism for event instantiation and textual support. The two signals are retained separately at decision time and jointly define the candidate-validity gate used before action selection.
- To correct argument extraction and assignment errors, we propose an action-gain-guided iterative correction strategy. Insertion, deletion, and split-and-reallocation are modeled as state-conditioned actions. Gain supervision is constructed from the change in document-

level matching score before and after each action, enabling the learned action-gain function to iteratively correct argument assignments.

2 Related Work

Related work falls into two broad categories. The first addresses event omissions, false-positive events, and argument misassignment in event extraction through recall enhancement, prediction correction, and structural verification. The second evaluates and iteratively optimizes structured outputs, providing the methodological basis for our gain formulation.

Recall Enhancement in Event Extraction. Existing studies have mainly addressed missed non-salient events through staged identification and schema enrichment. Staged methods first identify potential triggers or event types and subsequently perform type classification or grouped extraction, reducing the search difficulty associated with a large event-type space (Li et al., 2023; Liu et al., 2025). Another line of work enriches label semantics with type definitions, event schemas, and annotation guidelines to improve the recognition of fine-grained and non-salient event expressions (Xu et al., 2024; Srivastava et al., 2025). More recently, Jiang et al. (2026) formulate DEE as event-topology prediction and jointly model event- and role-aware representations for complete event-table construction. These methods strengthen holistic event construction, while our focus is complementary: recovering events outside the base extractor's proposal space and deciding whether state-dependent record edits improve the current structured output. Recent studies further exploit the semantic understanding and generation capabilities of LLMs to expand the event candidate space through task decomposition and prompt engineering (Shiri et al., 2024; Liu et al., 2025). Although candidate expansion improves event coverage, it also introduces more spurious and redundant candidates and aggravates conflicts in argument assignment across events. Candidate expansion can improve event extraction only when missed-event recovery, false-positive suppression, and argument-assignment correction are properly balanced.

Prediction Correction and Structural Verification in Event Extraction. Existing work reduces event errors through generation decomposition, post-hoc LLM correction, executable-schema verification, and constrained document-level generation (Shiri et al., 2024; Zhu et al., 2024; Guo et al., 2026; Du et al., 2022; Huang et al., 2023). DEEIA further models inter-event dependencies in one pass to extract arguments for multiple given events (Liu et al., 2024). These methods strengthen local or jointly encoded predictions, whereas our setting additionally contains missing and spurious event instances, requiring state-dependent edits over complete event records.

Gain-Guided Structured Refinement. Global information extraction has enforced document-level consistency through integer linear programming and dual decomposition (Yang & Mitchell, 2016), while learning-to-search, imitation-learning, and value-network methods optimize state-dependent decisions over partial structures (Daumé III et al., 2009; Chang et al., 2015; Chen et al., 2023; Gygli et al., 2017). We adopt this state-dependent view, restrict search to event-record edits, and supervise each action by its change in document-level matching score.

3 Method

3.1 Task Formulation and Framework Overview

Given a document $D = (x_1, \ldots, x_n)$ and an event schema $\Sigma = \{(y, \mathcal{R}_y) \mid y \in \mathcal{T}\}$, DEE aims to predict a set of event records $\widehat{\mathcal{Y}} = \{e_1, \ldots, e_M\}$. Each record $e_m = (y_m, p_m, A_m)$ contains an event type y_m, a trigger expression p_m, and a set of role–argument pairs $A_m = \{(r, a) \mid r \in \mathcal{R}_{y_m}\}$.

As shown in Figure 3, the base extractor $\mathcal{B}_\psi$ produces the initial event-record state S_0. The candidate event generator guided by discriminatively enhanced schemas produces an expanded candidate event set $\mathcal{C}^{\exp}$, which is deduplicated against S_0 to form the initial pending candidate event set:

$$S_0 = \mathcal{B}_\psi(D, \Sigma), \qquad \mathcal{C}_0 = \mathrm{Dedup}\left(\mathcal{C}^{\exp} \mid S_0\right). \tag{1}$$

At iteration t, the evaluator considers the current record state S_t and the pending candidate event set $\mathcal{C}_t$, estimates candidate-event eventness and state-conditioned action gains in parallel, and jointly

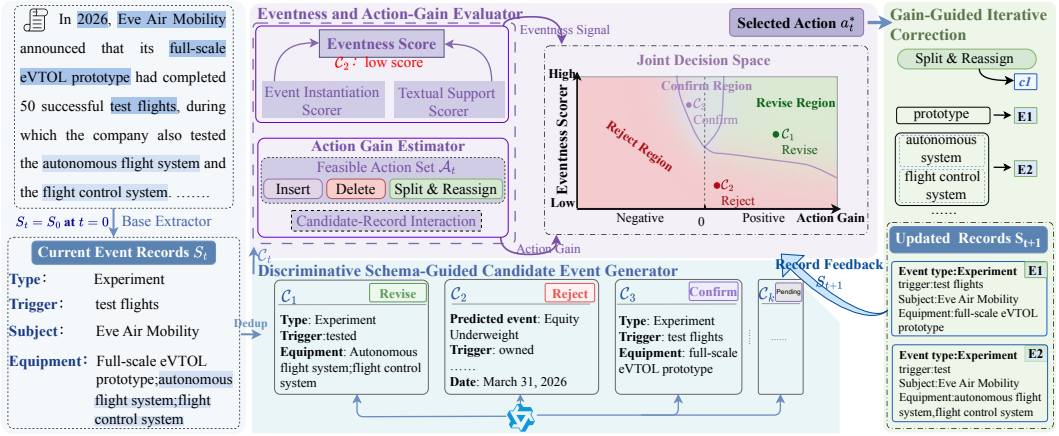

Figure 3: Gain-Guided Iterative Correction framework. The base extractor produces the initial event-record state S_0. Guided by discriminatively enhanced schemas, the candidate event generator supplements potentially missed events and deduplicates them against S_0 to form the initial pending candidate event set $\mathcal{C}_0$. At iteration t, the evaluator estimates candidate-event eventness and state-conditioned action gain in parallel, jointly assigns each candidate event to REJECT, CONFIRM, or REVISE, and executes the highest positive-gain action to update the event records. Action gains are recomputed after each update until no executable positive-gain action remains.

determines the state of each candidate event. The model executes the valid action with the highest predicted gain, updates the event records, and recomputes the remaining action gains until no executable positive-gain action remains.

3.2 DISCRIMINATIVE SCHEMA-GUIDED CANDIDATE EVENT GENERATOR

Given document D, event schema Σ, and enhanced schema prompt $\mathcal{P}_\Sigma$, a frozen LLM $\mathcal{G}_\phi$ generates $\mathcal{C}^{\text{exp}} = \mathcal{G}_\phi(D, \Sigma, \mathcal{P}_\Sigma)$. In addition to event-type definitions and role constraints, $\mathcal{P}_\Sigma$ incorporates event-instance boundaries and criteria for distinguishing similar events, derived from event definitions and annotation guidelines. These additions distinguish independent event instances from static or non-event descriptions and from semantically similar events. The enhanced schemas are generated offline by an LLM from existing event definitions and annotation guidelines and are manually checked against the annotation guidelines; no document-level annotations are used in this process. The same construction procedure transfers across domains by replacing the underlying schema descriptions and revising only domain-specific boundary and distinction criteria.

Each candidate event is represented as $c_i = (y_i, p_i, A_i, E_i)$, where E_i contains document-aligned trigger and argument evidence and defines the local context D_i. Candidates with unaligned triggers are discarded. Expansion is invoked once to fix the proposal pool and bound LLM cost; their action gains are recomputed after every state update.

3.3 EVENTNESS AND ACTION-GAIN EVALUATOR

The evaluator separately estimates the eventness of each candidate event and its state-conditioned action gains, and jointly uses the two scores at decision time. Action gains are dynamically updated with the current record state S_t.

Eventness Score. The *eventness score* jointly measures the degree of event instantiation and the level of textual support for a candidate event. For a candidate event $c_i \in \mathcal{C}_t$, the model encodes its local context, event structure, and event schema:

$$h_i^e = \text{Enc}_{\theta_e}(\text{Ser}_e(D_i, \Sigma, c_i)), \quad v_i = \sigma(w_v^\top h_i^e + b_v), \quad s_i = \sigma(w_s^\top h_i^e + b_s). \tag{2}$$

Here, D_i is the evidence-preserving context derived from E_i. v_i measures whether the text instantiates the target event type, whereas s_i measures whether the trigger expression and role–argument links are jointly supported by the same event instance. The validity gate retains the two factors separately: $v_i \geq \tau_e$ and $s_i \geq \tau_s$.

State-Conditioned Action Gain. The action-gain branch performs state-conditioned modeling independently of the eventness branch. For a candidate event c_i, the model retrieves records sharing its event type or a normalized role filler. Besides $\mathrm{INS}(i)$, $\mathrm{SPLIT}(i, j, b)$ pairs c_i with a related record e_j and enumerates MOVE/SHARE assignments for their matched fillers. Every matched filler enters the candidate record and is either removed from or retained in e_j. No-op, duplicate, ungrounded, and schema-invalid plans are pruned; construction and complexity details are provided in Appendix B.2. The model constructs $\mathrm{DEL}(j)$ for each eligible record $e_j \in S_t$. Together, these operations form the feasible action set $\mathcal{A}_t$ shown in Figure 3.

For each action $a \in \mathcal{A}_t$, the model jointly encodes its target $\xi(a)$, related records $\mathcal{N}^{(t)}(a)$, the document, the event schema, and the action type, and predicts its gain relative to the current state:

$$g_a^{(t)} = w_g^\top \mathrm{Enc}_{\theta_g}\left(\mathrm{Ser}_g\left(D, \Sigma, \xi(a), \mathcal{N}^{(t)}(a), a\right)\right) + b_g. \tag{3}$$

Here, $\xi(a)$ denotes either the candidate event involved in the action or the record to be deleted. Insertion and different split-and-reallocation plans are scored independently.

Joint Candidate–Action Decision. Let $\mathcal{A}_i^{(t)}$ denote the action set associated with candidate event c_i, and let its maximum action gain be $\bar{g}_i^{(t)} = \max_{a \in \mathcal{A}_i^{(t)}} g_a^{(t)}$. The evaluator jointly determines the candidate-event state from the event-instantiation score v_i, textual-support score s_i, and maximum action gain $\bar{g}_i^{(t)}$:

$$d_i^{(t)} = \begin{cases} \textsc{Reject}, & v_i < \tau_e \ \vee \ s_i < \tau_s, \\ \textsc{Confirm}, & v_i \geq \tau_e,\ s_i \geq \tau_s,\ \bar{g}_i^{(t)} \leq \tau_g, \\ \textsc{Revise}, & v_i \geq \tau_e,\ s_i \geq \tau_s,\ \bar{g}_i^{(t)} > \tau_g. \end{cases} \tag{4}$$

A candidate event with low eventness is assigned to REJECT. A candidate event with high eventness but no valid positive-gain action is assigned to CONFIRM, meaning that no state change is warranted for that candidate; this label does not by itself insert the candidate into the output. A high-eventness candidate with a positive-gain action is assigned to REVISE. In Figure 3, c_2 is rejected because of its low eventness. The event content of c_3 is already covered by the current records, so its associated actions yield no valid positive gain and it is confirmed. For c_1, the split-and-reallocation action recovers a missed event and corrects argument assignments, so the candidate event enters the revision region.

3.4 GAIN-GUIDED ITERATIVE CORRECTION

Let $\mathcal{A}_t^+ = \{a \in \mathcal{A}_t \mid g_a^{(t)} > \tau_g\}$ after eventness gating for candidate-conditioned actions. The model selects the action with the highest predicted gain and updates the event records:

$$a_t^* = \arg\max_{a \in \mathcal{A}_t^+} g_a^{(t)}, \qquad S_{t+1} = \mathcal{E}(S_t, a_t^*). \tag{5}$$

Ties are resolved by fewer modified role–argument pairs and then a fixed action identifier. Only one action is executed before the remaining gains are recomputed.

Rejected and confirmed candidates, together with candidates consumed by INS/SPLIT, are removed from the pending set; DEL only updates the record state. Remaining action gains are recomputed under S_{t+1}. When $\mathcal{A}_t^+ = \varnothing$, the iteration terminates and returns $\widehat{\mathcal{Y}} = S_T$.

3.5 TRAINING OBJECTIVES

Training jointly optimizes event-instantiation classification, textual-support classification, and action-gain estimation. Given the gold event-record set $\mathcal{Y}$, the ground-truth gain of action a is defined as the change in document-level matching score before and after applying the action:

$$\Delta_a^{(t)} = M_{\mathrm{doc}}(\mathcal{E}(S_t, a), \mathcal{Y}) - M_{\mathrm{doc}}(S_t, \mathcal{Y}), \quad M_{\mathrm{doc}}(S, \mathcal{Y}) = \frac{1}{2} F_{\mathrm{trg}}^{\mathrm{doc}}(S, \mathcal{Y}) + \frac{1}{2} F_{\mathrm{arg}}^{\mathrm{doc}}(S, \mathcal{Y}). \tag{6}$$

A positive value $\Delta_a^{(t)} > 0$ indicates that the action improves the current extraction result. The regressor is used for gain-sign filtering and within-state ranking, rather than calibrated prediction

of the exact metric change. Training states are constructed through event deletion, record merging, record duplication, and spurious-event insertion, which respectively simulate missed events, argument attachment to mixed records, redundant predictions, and erroneous event records. The event-instantiation label is $y_i^v = 1$ when the event type and trigger expression of a candidate event correspond to a gold event instance. The textual-support label is $y_i^s = 1$ when its trigger expression and role–argument links are jointly supported by the same gold event instance. All remaining labels are 0.

For batch sets $\mathcal{Q}$ and $\mathcal{A}$, the training objective is

$$\mathcal{L} = \frac{1}{|\mathcal{Q}|} \sum_{c_i \in \mathcal{Q}} [\mathrm{BCE}(v_i, y_i^v) + \lambda_s \, \mathrm{BCE}(s_i, y_i^s)] + \frac{\lambda_g}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \left(g_a^{(t)} - \Delta_a^{(t)} \right)^2 . \tag{7}$$

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on FNDEE, a Chinese military-news dataset with multiple events, nested entities, and long-span arguments (Xu et al., 2024); CrudeOilNews, English market news covering macroeconomic and geopolitical developments (Lee et al., 2022); and SENTiVENT, English economic news with fine-grained corporate events and cross-sentence arguments (Jacobs & Hoste, 2022).

Baselines. FNDEE baselines are BERT-CRF, EEQA, MLBiNet, EDM3, and SEELE (Xu et al., 2024); CrudeOilNews and SENTiVENT baselines are DyGIE++, OneIE, adapted Guideline-Prompt (Srivastava et al., 2025), adapted SEELE, and an LLM directly prompted with the event schema. †/‡ denote reported/reproduced-or-adapted results.

Evaluation Metrics. For FNDEE, we report P/R/F_1 for trigger and argument extraction, and count missed triggers, spurious triggers, missing arguments, over-extracted arguments, and role errors. For CrudeOilNews and SENTiVENT, we report P/R/F_1 for trigger classification (Trig-C) and argument classification (Arg-C). Derived metrics in the mechanism analyses are defined in the corresponding table captions or analysis paragraphs.

Implementation Details. SEELE (Xu et al., 2024) serves as Base, reproduced on FNDEE under the original protocol and adapted to the event-type and role schemas of the two English datasets. All FNDEE analyses use this reproduction, whose decoded records define S_0.

Candidates are generated by schema-guided Qwen2.5-7B-Instruct. The eventness and action-gain encoders use `hfl/chinese-roberta-wwm-ext` for FNDEE and `roberta-large` for the English datasets; event instantiation and textual support share the eventness encoder, while the action-gain encoder has independent parameters. We use a maximum input length of 500, train/evaluation batch sizes of 8/16, and five AdamW epochs at 3×10^{-5}. Thresholds are selected on the development set and fixed; state perturbations and label transformations are training-only. Official splits and evaluation scripts are used. Ours denotes the full framework initialized with S_0 and $\mathcal{C}_0$.

4.2 Main Results

Overall Results. Tables 1 and 2 show a consistent recall improvement after candidate expansion and structured refinement. Relative to the corresponding SEELE base extractor, BALANCE-DEE improves trigger/argument F_1 from 81.6/67.0 to 82.7/68.7 on FNDEE, from 61.51/49.10 to 62.12/50.75 on CrudeOilNews, and from 47.10/42.79 to 50.38/45.69 on SENTiVENT. The largest gains occur on SENTiVENT, while CrudeOilNews exhibits a clearer

Table 1: Main results on FNDEE.

Model	Trigger			Argument		
	P	R	F1	P	R	F1
BERT-CRF	**83.8**†	68.2†	75.2†	–	–	–
EEQA	77.4†	80.1†	78.7†	–	–	–
MLBiNet	74.0†	77.4†	75.7†	–	–	–
EDM3	73.6†	80.0†	76.5†	–	–	–
Base (SEELE)	76.7‡	87.1‡	81.6‡	65.8‡	68.2‡	67.0‡
Ours	77.0‡	**89.4**‡	**82.7**‡	**68.9**‡	**68.4**‡	**68.7**‡

precision–recall trade-off: trigger recall increases from 58.70 to 63.40 with a reduction in precision. Together with the error and candidate-coverage analyses below, these results suggest that the framework primarily improves coverage while its eventness and gain components recover part of the precision lost during candidate expansion. Table 9 further shows that BALANCE-DEE improves both trigger and argument F_1 when initialized from DyGIE++, suggesting that its refinement benefit is not unique to SEELE.

Table 2: Main results on CrudeOilNews and SENTiVENT. †: reported; ‡: reproduced/adapted. Best/second-best results are bold/underlined.

Model	CrudeOilNews			SENTiVENT		
	P	R	F1	P	R	F1
Trigger classification (Trig-C)						
DyGIE++‡	**75.62**	53.96	**62.98**	38.86	46.81	42.46
OneIE‡	51.86	37.12	43.27	**61.51**	34.10	43.88
Guideline-Prompt (adapted)‡	<u>64.80</u>	54.20	59.03	<u>50.20</u>	44.10	46.95
SEELE‡	64.60	<u>58.70</u>	61.51	43.60	<u>51.20</u>	<u>47.10</u>
LLM-Prompt‡	15.20	20.80	17.56	21.59	5.80	9.15
Ours	60.90	**63.40**	<u>62.12</u>	46.20	**55.40**	**50.38**
Argument classification (Arg-C)						
DyGIE++‡	**62.72**	28.79	39.46	38.95	37.12	38.01
OneIE‡	48.51	32.45	38.89	35.00	28.00	31.11
Guideline-Prompt (adapted)‡	50.90	39.60	44.54	**44.80**	38.70	41.53
SEELE‡	<u>53.60</u>	<u>45.30</u>	<u>49.10</u>	40.30	<u>45.60</u>	<u>42.79</u>
LLM-Prompt‡	8.60	11.90	9.98	6.48	2.00	3.06
Ours	51.00	**50.50**	**50.75**	<u>42.20</u>	**49.80**	**45.69**

4.3 Ablation Study

Table 3: Ablation results on FNDEE. Bold and underlined values indicate the best and second-best results.

Model	Trigger			Argument		
	P	R	F1	P	R	F1
Full Model	<u>77.00</u>	89.40	**82.74**	<u>68.90</u>	68.40	**68.65**
w/o Candidate Expansion	**79.00**	84.80	81.80	**69.80**	66.20	<u>67.95</u>
w/o Schema Guidance	73.00	88.40	79.97	65.60	66.80	66.19
w/o Eventness	73.80	<u>90.40</u>	81.26	64.90	69.50	67.12
w/o Action Gain	69.80	**91.00**	79.00	61.80	**70.40**	65.82
w/o Record Context	72.70	89.80	80.35	64.10	<u>69.80</u>	66.83
w/o Action-Specific Gain	76.10	88.60	<u>81.88</u>	67.20	67.80	67.50

Component Contributions. Table 3 shows that candidate event expansion and schema guidance determine the coverage ceiling and effective density of the candidate event space, respectively. Removing candidate event expansion reduces trigger and argument recall by 4.6 and 2.2 percentage points, whereas removing schema guidance substantially degrades precision and F_1. Removing eventness slightly increases recall but causes a pronounced precision loss. Removing action gain produces the most severe overall degradation, reducing trigger and argument precision by 7.2 and 7.1 percentage points. The results without record context or action-specific modeling further show that action gain must be estimated with respect to the current record state and the specific operation.

4.4 Mechanism Analysis

Error-Type Analysis. Table 4 shows that the full model reduces missed triggers by 17.8%, with only a 0.9% increase in spurious triggers, indicating that recovery gains substantially outweigh the added false-positive cost. It also reduces over-extracted arguments by 13.2% and total errors

by 6.8%. Table 5(a) further shows that the cross-event argument attachment rate decreases from 20.4% to 12.9%, confirming the balance between missed-event recovery, false-positive control, and argument correction.

Table 4: Error analysis on FNDEE relative to S_0. Negative values indicate fewer errors.

Model	Trigger		Argument			Total Err.
	Missed Trig.	Spurious Trig.	Missing Arg.	Extra Arg.	Wrong Role	
S_0	572	1,174	3,792	4,242	81	**9,861**
Ours	470	1,184	3,775	3,680	77	**9,186**
Rel. change	$\mathbf{-17.8\%}$	$+0.9\%$	-0.4%	$\mathbf{-13.2\%}$	-4.9%	$\mathbf{-6.8\%}$

Complementarity of Eventness and Action Gain. Figure 4(a) shows that using eventness alone achieves 91.0% recall but only 69.8% precision, while using action gain alone also fails to attain the performance balance of the full model. Jointly modeling eventness and state-conditioned action gain yields 77.0% precision and 89.4% recall, achieving the highest overall F_1. The degradation without record state confirms the need for state-conditioned gain, while Figure 4(b) shows stable performance around the selected thresholds $(0.26, 0.28)$.

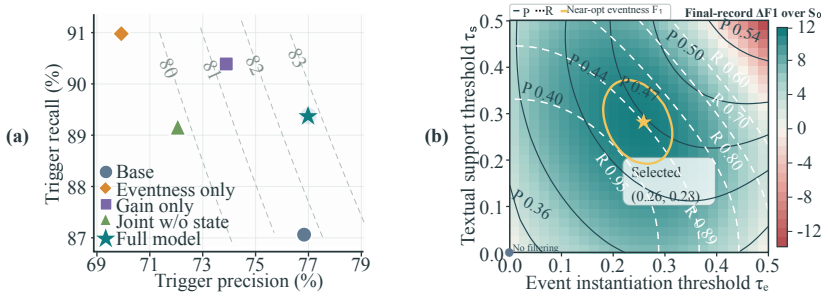

Figure 4: Joint-decision analysis on FNDEE: (a) final-record precision and recall under different configurations; (b) effects of eventness thresholds on final-record F_1 gain.

Event Competition and Salience. Events in the same document are considered competing when they share an event type and a role-filler value, and are divided into initial and later events according to trigger position. Because FNDEE has no human event-salience labels, we use a fixed post-hoc textual-salience proxy that combines trigger position, mention frequency, and discourse centrality. The proxy is computed without test-time tuning and is used only for analysis; its full definition and limitations are provided in Appendix B.1.

Table 5(a) shows that recall for later competing events increases by 7.2 percentage points, while their argument attachment rate decreases from 20.4% to 12.9%. Recall for the non-salient subset increases by 4.2 percentage points, compared with only 0.4 points for the salient subset. The improvements are concentrated on event subsets for which the base model is prone to omissions or argument confusion.

Coverage of Candidate Event Expansion. Schema prompting raises missed-event proposal coverage from 31.68% to 72.90% with fewer candidates. The fused pool covers 98.40% of gold events under the proposal criterion, while exact trigger recall improves from 87.1% to 89.4% after refinement.

Eventness Evaluation on a Fixed Candidate Event Pool. Table 5(c) shows that jointly modeling event instantiation and textual support achieves the highest candidate event retention F_1 and spurious-event rejection rate, while retaining 89.0% of true candidate events. Zero-shot NLI degenerates to retaining all candidate events, indicating that its scores cannot form an effective decision boundary.

Action and Iteration Analysis. Table 6 shows that enumeration raises positive-plan coverage from 61.4% to 84.9%, while gain ranking selects a positive plan in 70.4% of covered cases, yielding 59.8% overall Pos@1. Iterative correction further improves F_1 from 82.3 to 82.7 and reduces mixed-record errors from 13.1% to 9.6%. The remaining gap to oracle gain suggests that action ranking still constitutes an important source of residual error.

Table 5: FNDEE diagnostics. In (a), R denotes exact trigger recall and C denotes proposal coverage. In (b), Gold Cov. uses one-to-one matching by event type and overlapping trigger spans; Miss Rec. measures coverage of gold events missed by Base. In (c), K-P/K-R/K-F1 are retention scores and P-Rej. is spurious-event rejection.

(a) Event subsets and recovery

Subset	#Gold	Base R	Fusion C	Ours R	ΔR	Attach.↓
All	4,435	87.1	98.4	89.4	+2.3	–
Later (comp.)	1,178	78.4	97.3	85.6	**+7.2**	20.4→12.9
Initial (comp.)	1,100	91.2	99.0	91.6	+0.4	–
Non-comp.	2,157	89.8	98.7	90.4	+0.6	–
Non-salient (proxy)	2,218	82.1	97.5	86.3	**+4.2**	–
Salient (proxy)	2,217	92.1	99.3	92.5	+0.4	–

(b) Candidate-event proposal coverage

	Base	Qwen		Fusion	
		label	schema	label	schema
#Candidate Events	5,594	11,400	10,580	15,742	14,386
/Doc	2.24	4.56	4.23	6.30	5.75
Gold Cov.	4,173 (94.09)	2,598 (58.58)	3,668 (82.71)	4,256 (95.96)	**4,364 (98.40)**
Miss Rec.	–	31.68	**72.90**	–	–

(c) Eventness on a fixed candidate pool

	All	NLI	Single	Qwen	Inst.	Supp.	**Joint**
Thr.	–	$(0.00, 0.00)$	0.18	Prompt	0.36	0.44	$(0.26, 0.28)$
K-P	0.329	0.329	0.438	0.397	0.410	0.390	**0.468**
K-R	1.000	1.000	0.856	0.914	0.897	0.894	0.890
K-F1	0.495	0.495	0.579	0.554	0.563	0.543	**0.613**
P-Rej.	0.000	0.000	0.462	0.319	0.367	0.317	**0.505**

Table 6: Action-level diagnostics on FNDEE: (a) gain estimation, (b) SPLIT search, and (c) one-shot vs. iterative correction. Act./Doc and Mixed Err. denote actions/document and mixed-record error rate.

(a) Action-gain estimation

Action	BAcc.	F_1^+	Pos@1	r	ρ
Overall	72.4	66.9	63.2	0.45	0.42
INS	**73.8**	**68.2**	**65.1**	**0.48**	**0.45**
SPLIT	70.7	64.6	59.8	0.39	0.36
DEL	73.1	67.5	64.0	0.46	0.43

(b) SPLIT search diagnosis

Strategy	Plans	Cov.$^+$	Sel.$^+$ | Cov$^+$	Arg. F_1	Mix. Err.
w/o SPLIT	0.00	–	–	67.74	14.8%
Move-all	1.00	61.4	–	68.02	12.7%
Enumerated	4.27	84.9	70.4	**68.65**	**9.6%**
Oracle gain	4.27	84.9	100.0	69.41	6.7%

(c) One-shot selection versus iterative correction

Strategy	Act./Doc	P	R	F1	Mixed Err.
One-Shot Selection	0.93	76.6	88.9	82.3	13.1%
Iterative Correction	1.47	**77.0**	**89.4**	**82.7**	**9.6%**

5 CONCLUSION

We propose BALANCE-DEE, a document-level event extraction framework that reconciles candidate expansion with false-positive correction through gain-guided record revision. Experiments on three datasets demonstrate consistent improvements in event recall, with particularly strong gains for non-salient and competing events, while reducing over-extracted arguments and mixed records. Performance remains constrained by the coverage of the initial proposal pool and the reliability of cross-domain and SPLIT gain estimation. The textual-salience analysis also relies on a fixed proxy rather than human salience labels, so the corresponding result should be interpreted as evidence about non-salient event expressions rather than a universal notion of event importance. These limitations motivate future work on state-conditioned re-proposal, stronger action ranking, and human-validated salience evaluation.

AI USE STATEMENT

Generative AI tools were used to assist with literature retrieval and discovery, research brainstorming, limited aspects of experimental design and methodological feedback, drafting and revising portions of the manuscript, translation, language polishing, and interpretation of experimental results. All AI-assisted suggestions, technical claims, citations, and experimental interpretations were reviewed and verified by the authors before inclusion. The authors take full responsibility for the final methodology, experiments, analysis, and presentation of this work.

REPRODUCIBILITY STATEMENT

The main paper specifies the task formulation, candidate generation procedure, eventness and action-gain models, action set, training objective, datasets, evaluation metrics, and core optimization settings. Appendix B.1 defines the textual-salience proxy used only for post-hoc analysis, Appendix B.2 gives the evidence alignment, neighborhood construction, split-and-reallocation search space, and complexity bound, and Appendix B.3 reports gain-threshold sensitivity and stage-wise inference cost.

REFERENCES

Samuel Bayer, Cheryl Clark, Oanh Dang, John Aberdeen, Sonja Brajovic, Kimberley Swank, Lynette Hirschman, and Robert Ball. ADE Eval: An evaluation of text processing systems for adverse event extraction from drug labels for pharmacovigilance. *Drug Safety*, 44:83–94, 2021. doi: 10.1007/s40264-020-00996-3.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, and John Langford. Learning to search better than your teacher. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2058–2066. PMLR, 2015. URL https://proceedings.mlr.press/v37/changb15.html.

Yunmo Chen, William Gantt, Weiwei Gu, Tongfei Chen, Aaron Steven White, and Benjamin Van Durme. Iterative document-level information extraction via imitation learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1858–1874, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.136. URL https://aclanthology.org/2023.eacl-main.136/.

Hal Daumé III, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009. doi: 10.1007/s10994-009-5106-x.

Xinya Du, Sha Li, and Heng Ji. Dynamic global memory for document-level argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5264–5275, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.361. URL https://aclanthology.org/2022.acl-long.361/.

Quanjiang Guo, Sijie Wang, Jinchuan Zhang, Ben Zhang, Zhao Kang, Ling Tian, and Ke Yan. Extracting events like code: A multi-agent programming framework for zero-shot event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30880–30887, 2026. doi: 10.1609/aaai.v40i37.40346. URL https://doi.org/10.1609/aaai.v40i37.40346.

Michael Gygli, Mohammad Norouzi, and Anelia Angelova. Deep value networks learn to evaluate and iteratively refine structured outputs. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1341–1351. PMLR, 2017.

Fatemeh Haji, Mazal Bethany, Cho-Yu Jason Chiang, Anthony Rios, and Peyman Najafirad. Reflective agreement: Combining self-mixture of agents with a sequence tagger for robust event

extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30476–30492, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1550. URL https://aclanthology.org/2025.emnlp-main.1550/.

Jianpeng Hu, Chao Xue, Chunqing Yu, JiaCheng Xu, and Chengxiang Tan. Joint learning event-specific probe and argument library with differential optimization for document-level multi-event extraction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 714–726, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.42. URL https://aclanthology.org/2025.findings-naacl.42/.

Guanhua Huang, Runxin Xu, Ying Zeng, Jiaze Chen, Zhouwang Yang, and Weinan E. An iteratively parallel generation method with the pre-filling strategy for document-level event extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10834–10852, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.668. URL https://aclanthology.org/2023.emnlp-main.668/.

Gilles Jacobs and Véronique Hoste. SENTiVENT: Enabling supervised information extraction of company-specific events in economic and financial news. *Language Resources and Evaluation*, 56(1):225–257, 2022. doi: 10.1007/s10579-021-09562-4. URL https://doi.org/10.1007/s10579-021-09562-4.

Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pp. 254–262, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL https://aclanthology.org/P08-1030/.

Bin Jiang, Sendong Zhu, Junyi Wu, and Yu Liu. A joint learning framework for document-level event extraction. *IEEE Transactions on Neural Networks and Learning Systems*, 2026. doi: 10.1109/TNNLS.2026.3678139.

Disha Jindal, Daniel Deutsch, and Dan Roth. Is killed more significant than fled? a contextual model for salient event detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 114–124, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.10. URL https://aclanthology.org/2020.coling-main.10/.

Meisin Lee, Lay-Ki Soon, Eu Gene Siew, and Ly Fie Sugianto. CrudeOilNews: An annotated crude oil news corpus for event extraction. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 465–479, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.lrec-1.49/.

Sha Li, Qiusi Zhan, Kathryn Conger, Martha Palmer, Heng Ji, and Jiawei Han. GLEN: General-purpose event detection for thousands of types. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2823–2838, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.170. URL https://aclanthology.org/2023.emnlp-main.170/.

Wanlong Liu, Li Zhou, DingYi Zeng, Yichen Xiao, Shaohuan Cheng, Chen Zhang, Grandee Lee, Malu Zhang, and Wenyu Chen. Beyond single-event extraction: Towards efficient document-level multi-event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 9470–9487, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.564. URL https://aclanthology.org/2024.findings-acl.564/.

Wenxuan Liu, Zixuan Li, Long Bai, Yuxin Zuo, Daozhu Xu, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. Towards event extraction with massive types: LLM-based collaborative annotation and partitioning extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34377–34399, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1743. URL https://aclanthology.org/2025.emnlp-main.1743/.

Zhengzhong Liu, Chenyan Xiong, Teruko Mitamura, and Eduard Hovy. Automatic event salience identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1226–1236, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1154. URL https://aclanthology.org/D18-1154/.

Fatemeh Shiri, Farhad Moghimifar, Reza Haffari, Yuan-Fang Li, Van Nguyen, and John Yoo. Decompose, enrich, and extract! schema-aware event extraction using LLMs. In *Proceedings of the 27th International Conference on Information Fusion (FUSION)*, pp. 1–8, Venice, Italy, 2024. IEEE. doi: 10.23919/FUSION59988.2024.10706385. URL https://doi.org/10.23919/FUSION59988.2024.10706385.

Kai Shuang, Ji Zhou, Qiwei Wang, and Jinyu Guo. Thinking about how to extract: Energizing LLMs' emergence capabilities for document-level event argument extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5520–5532, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.328. URL https://aclanthology.org/2024.findings-acl.328/.

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. Instruction-tuning LLMs for event extraction with annotation guidelines. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13055–13071, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.677. URL https://aclanthology.org/2025.findings-acl.677/.

Xinyu Wang, Lin Gui, and Yulan He. Document-level multi-event extraction with event proxy nodes and hausdorff distance minimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10118–10133, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.563. URL https://aclanthology.org/2023.acl-long.563/.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3533–3546, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.274. URL https://aclanthology.org/2021.acl-long.274/.

Zijie Xu, Peng Wang, Wenjun Ke, Guozheng Li, Jiajun Liu, Ke Ji, Xiye Chen, and Chenxiao Wu. Incorporating schema-aware description into document-level event extraction. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 6597–6605. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/729. URL https://www.ijcai.org/proceedings/2024/729.

Bishan Yang and Tom M. Mitchell. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 289–299, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1033. URL https://aclanthology.org/N16-1033/.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6298–6308, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.492. URL https://aclanthology.org/2021.acl-long.492/.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. Doc2EDAG: An end-to-end document-level framework for chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 337–346, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1032. URL https://aclanthology.org/D19-1032/.

Mengna Zhu, Kaisheng Zeng, Jibing Wu, Lihua Liu, Hongbin Huang, Lei Hou, and Juanzi Li. LC4EE: LLMs as good corrector for event extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12028–12038, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.715. URL https://aclanthology.org/2024.findings-acl.715/.

A Additional Experimental Analyses

A.1 Robustness to Gain Supervision and Transfer

Robustness to Gain Supervision. The default gain target is $M_\alpha = \alpha F_{\text{trg}}^{\text{doc}} + (1-\alpha) F_{\text{arg}}^{\text{doc}}$ with $\alpha = 0.5$. Table 7 varies α and also replaces the target with normalized error-count reduction. Final trigger and argument F_1 vary by at most 0.21 and 0.24 points, respectively, indicating that the final decisions are stable across these alternative gain targets.

Table 7: Robustness to alternative gain targets on FNDEE.

Target	BAcc.	F_1^+	Pos@1	r	Trig. F_1	Arg. F_1
$M_{0.25}$	71.9	66.5	62.7	0.44	82.63	68.72
$M_{0.50}$	72.4	66.9	63.2	0.45	82.74	68.65
$M_{0.75}$	72.2	66.7	63.0	0.44	82.79	68.48
Error reduction	71.7	66.1	62.5	0.42	82.58	68.54

Gain Threshold and Cross-Dataset Transfer. Lower τ_g executes more uncertain edits, whereas larger values reduce the number of actions. The selected $\tau_g = 0.01$ lies in a stable region: across $\tau_g \in \{0, 0.005, 0.01, 0.02, 0.04\}$, trigger and argument F_1 vary within 0.23 and 0.22 points, respectively. Transferring the full (τ_e, τ_s, τ_g) tuple between CrudeOilNews and SENTiVENT decreases trigger and argument F_1 by at most 0.29 and 0.32 points, respectively; fixing $\tau_g = 0$ changes them by at most 0.16 and 0.18 points. Additional gain-threshold sensitivity and runtime results on FNDEE are reported in Appendix B.3.

Runtime and Cost. Detailed stage-wise latency, memory, and LLM-call measurements are reported in Appendix B.3.

A.2 Effects of Schema Specification

We compare event-type labels only, a vanilla schema with conventional event and role descriptions, and our enhanced schema with instance-boundary and similar-event distinction criteria on FNDEE. All other refinement stages remain fixed. Candidate counts refer to Qwen proposals before fusion with the base records. Missed-event recovery measures the fraction of events missed by the base extractor that occur among these proposals; final F_1 is computed after refinement.

Table 8: Schema specification on FNDEE. Miss Rec. is missed-event recovery (%).

Schema setting	#Candidates	Miss Rec.	Trig. F_1	Arg. F_1
Label only	11,400	31.68	79.97	66.19
Vanilla schema	10,960	54.20	81.68	67.44
Enhanced schema	10,580	72.90	**82.74**	**68.65**

Table 8 extends the label-only versus enhanced comparison in Table 5(b) with a vanilla-schema intermediate. Relative to the vanilla schema, the enhanced schema generates 380 fewer proposals but recovers 18.70 percentage points more events missed by the base extractor; final trigger and argument F_1 increase by 1.06 and 1.21 points. This combination suggests a higher concentration of recoverable events in the proposal pool, although candidate count alone does not determine its false-positive rate. The enhanced row agrees with Table 3; the label-only and enhanced proposal counts and recovery rates agree with Table 5(b).

A.3 Alternative Base Extractor on SENTiVENT

We initialize BALANCE-DEE with DyGIE++ records on SENTiVENT to assess its applicability beyond SEELE. The expansion-only configuration adds candidate proposals to DyGIE++ predictions before eventness verification and gain-guided correction; the full configuration applies both modules. We use the classification metrics of Section 4.2. Expansion alone raises trigger and argument recall by 8.09 and 7.48 points relative to DyGIE++, while precision falls by 4.26 and 5.15 points. Trigger F_1 stays essentially level (42.46 versus 42.45); argument F_1 increases by 0.45 points. The full framework raises precision by 5.60 and 6.50 points relative to expansion alone, with recall of 51.70 and 40.80, and reaches 45.23 trigger and 40.55 argument F_1. Relative to DyGIE++ itself, these are gains of 2.77 and 2.54 points. This supports use with a second base extractor on SENTiVENT; broader cross-dataset transfer remains untested.

Table 9: DyGIE++ initialization on SENTiVENT.

	Trigger			Argument		
Configuration	P	R	F_1	P	R	F_1
DyGIE++	38.86	46.81	42.46	38.95	37.12	38.01
+ Candidate Expansion	34.60	54.90	42.45	33.80	44.60	38.46
+ BALANCE-DEE	**40.20**	51.70	**45.23**	**40.30**	40.80	**40.55**

A.4 Variation Across Random Seeds

Table 10 reports mean $\pm$ sample standard deviation over five runs with different random seeds $\{13, 21, 42, 87, 100\}$. All runs use identical data splits, hyperparameters, and model-selection criteria. The means are close to the single-run entries in Tables 1 and 2; differences of a few hundredths reflect aggregation across runs. Across the five runs, mean trigger/argument F_1 gains are 1.12/1.63 points on FNDEE, 0.60/1.62 on CrudeOilNews, and 3.24/2.86 on SENTiVENT. All six mean differences favor BALANCE-DEE. The within-model standard deviation ranges from 0.18 to 0.36 points. These aggregate statistics do not constitute a paired significance test or imply that every individual seed improves.

Table 10: Multi-seed trigger and argument F_1 (mean $\pm$ sample standard deviation over five runs).

Dataset	Model	Trig. F_1	Arg. F_1
FNDEE	SEELE	81.60 ± 0.22	67.00 ± 0.28
	BALANCE-DEE	$\mathbf{82.72 \pm 0.18}$	$\mathbf{68.63 \pm 0.24}$
CrudeOilNews	SEELE	61.50 ± 0.27	49.11 ± 0.31
	BALANCE-DEE	$\mathbf{62.10 \pm 0.29}$	$\mathbf{50.73 \pm 0.26}$
SENTiVENT	SEELE	47.12 ± 0.34	42.81 ± 0.36
	BALANCE-DEE	$\mathbf{50.36 \pm 0.25}$	$\mathbf{45.67 \pm 0.29}$

A.5 Sensitivity to Candidate Expansion Strength

We evaluate four expansion scales in terms of candidate recall, spurious rate, and downstream F_1. The expansion-only configuration uses the candidate pool at each scale without eventness verification or gain-guided correction. BALANCE-DEE applies the full refinement framework to the same candidate pool. From $1.0\times$ to $3.0\times$, candidate recall rises by 11.7 points and the spurious rate by

Table 11: Sensitivity to candidate-expansion strength on FNDEE. Candidate recall and spurious rate are percentages.

Scale	Cand. recall	Spurious rate	Expansion-only F_1	BALANCE-DEE F_1
$1.0\times$	86.4	8.5	67.0	68.1
$1.5\times$	92.7	15.8	66.8	68.7
$2.0\times$	96.3	24.6	65.9	**68.9**
$3.0\times$	98.1	37.5	63.7	68.5

29.0 points. Expansion-only F_1 decreases from 67.0 to 63.7. BALANCE-DEE reaches a maximum of 68.9 at $2.0\times$ and remains at 68.5 at $3.0\times$. Correction moderates the loss under heavier expansion, but the decline at the largest scale shows that additional candidate coverage need not improve the final prediction. This observation is confined to the tested expansion range.

B Additional Method Details and Analyses

B.1 Event Salience Partition

FNDEE provides no event-salience labels. For post-hoc analysis only, we define a textual-salience proxy from trigger position, mention frequency, and discourse centrality; it is excluded from training, threshold selection, and inference. For a gold event $e = (y, p, A)$, $q_{\mathrm{pos}}(e)$ assigns larger values to earlier trigger sentences, $q_{\mathrm{freq}}(e)$ counts normalized trigger occurrences, and $q_{\mathrm{cent}}(e)$ is the cosine similarity between the character-level TF–IDF vector of the trigger sentence and the document centroid. Each signal is converted to a training-set percentile rank:

$$\widetilde{q}_k(e) = \frac{1}{|\mathcal{E}_{\mathrm{tr}}|} \sum_{e' \in \mathcal{E}_{\mathrm{tr}}} \mathbb{I}[q_k(e') \le q_k(e)], \qquad k \in \{\mathrm{pos}, \mathrm{freq}, \mathrm{cent}\}. \tag{8}$$

We then compute

$$\mathrm{Sal}(e) = \frac{\widetilde{q}_{\mathrm{pos}}(e) + \widetilde{q}_{\mathrm{freq}}(e) + \widetilde{q}_{\mathrm{cent}}(e)}{3}. \tag{9}$$

The training-set median τ_{sal} partitions test events into non-salient and salient proxy groups. The fixed, untuned proxy reflects observable textual prominence and is not equivalent to human-annotated event salience.

B.2 Action Construction and Search

Evidence and Serialization. Candidate triggers and arguments are aligned to document spans after whitespace, case, and punctuation normalization. Candidates with unaligned triggers are discarded, whereas unaligned arguments are removed. E_i stores aligned spans and sentence indices, and D_i is their minimum covering sentence interval. Eventness inputs serialize D_i, the schema, and the candidate; gain inputs additionally serialize the action, related records, and reallocation plan. Roles follow schema order, whereas fillers and records follow textual order.

Neighborhood and Feasibility. For candidate c_i and state S_t, we retrieve

$$\mathcal{N}_i^{(t)} = \{e_j \in S_t \mid y_j = y_i \ \vee \ \exists a \in A_i, a' \in A_j : \mathrm{Match}(a, a') = 1\}, \tag{10}$$

where Match denotes identical aligned spans or normalized surface forms. Actions must preserve document grounding and type–role compatibility, change the state, and avoid duplicate records.

Split-and-Reallocation Plans. For candidate c_i and related record e_j, let

$$\mathcal{U}_{ij} = \{a' \in A_j \mid \exists a \in A_i : \mathrm{Match}(a, a') = 1\} \tag{11}$$

contain their shared fillers. Each $u \in \mathcal{U}_{ij}$ is assigned MOVE, which removes it from e_j, or SHARE, which retains it in both records. The candidate defines the new record, and the updated old record is

$$e_j^b = (y_j, p_j, A_j \setminus \{u \in \mathcal{U}_{ij} : b(u) = \textsc{Move}\}). \tag{12}$$

The all-SHARE plan is removed because it is equivalent to insertion. Let $N_C^{(t)} = |\mathcal{C}_t|$, $N_R^{(t)} = |S_t|$, $\bar{n}_t$ denote the average neighborhood size, and $K_t = \max_{i,j} |\mathcal{U}_{ij}|$. Then

$$|\mathcal{A}_t| \le N_C^{(t)} + N_R^{(t)} + N_C^{(t)} \bar{n}_t (2^{K_t} - 1). \tag{13}$$

One action is executed per iteration and the remaining gains are recomputed. Ties are resolved by fewer modified role–argument pairs and then a fixed action identifier. On FNDEE, each candidate retrieves 1.68 records on average and each candidate–record pair yields 4.27 feasible split plans; the maximum number of matched fillers is 7.

B.3 GAIN-THRESHOLD SENSITIVITY AND RUNTIME

Table 12: Sensitivity to the gain threshold on FNDEE.

τ_g	Trig. F_1	Arg. F_1	Act./Doc
0.000	82.66	68.60	1.59
0.005	82.67	68.60	1.53
0.010	82.74	68.65	1.47
0.020	82.65	68.59	1.34
0.040	82.51	68.43	1.12

Table 13: Stage-wise inference cost on FNDEE. Runtime is measured with batch size one on a single RTX 4090 after 20 warm-up documents and averaged over three runs; model loading is excluded. Candidate generation uses one local Qwen2.5-7B-Instruct call per document, averaging 1,642 input and 238 output tokens.

Stage	LLM Calls	Mean (s)	P95 (s)	Peak GB
Base extractor	0	0.31	0.53	4.8
Candidate generation	1	4.41	7.82	17.8
Eventness scoring	0	0.08	0.14	4.2
Gain scoring/search	0	0.22	0.41	5.1
Full pipeline	1	5.02	8.63	17.8