

EventKD: Event-Aware Knowledge Distillation for Generative Event Extraction

Hoang Thanh Dat^{1*}, Quoc Phong Dao^{1*}, Nam Le Hai^{1†}, Franck Dernoncourt²,
Linh Ngo Van¹, Trung Le³, Thien Huu Nguyen⁴

¹Hanoi University of Science and Technology, ²Adobe Research,

³Monash University, ⁴University of Oregon

Abstract

Event extraction aims to identify triggers in text, classify them into predefined event types, and recognize associated arguments such as participants and locations. Large language model (LLM)-based generative approaches provide a unified paradigm for joint trigger identification, classification, and argument extraction; however, their computational cost necessitates knowledge distillation (KD) into smaller student models. Existing KD methods for LLM distillation operate at the token level, treating each output position independently and thus failing to concentrate the distillation signal on event-relevant spans or to capture the relational structure among event components. We propose **EventKD**, a two-level distillation framework tailored for event extraction. At the token level, a distributional divergence loss incorporates event-aware span importance weighting, derived from the teacher’s structured output, to prioritize tokens corresponding to triggers, arguments, event type information. At the span level, an Event-Aware (EA) loss aligns pairwise distance structures among event-aware span representations between teacher and student across multiple Transformer layers. Experiments on three benchmarks (ACE05, GENEVA and MAVEN) demonstrate that EventKD consistently outperforms existing distillation methods, e.g., improving trigger F1 by 2.18 points and argument F1 by 5.32 points on ACE05 over the strongest baselines. Additional experiments under a different LLM family (Llama-3) and a non-English benchmark (MINION Spanish) further confirm that these gains generalize beyond the original Qwen3 English setup.

1 Introduction

Event Extraction (EE) is a core task in information extraction concerned with identifying event triggers in text, classifying them into predefined event

types, and recognizing the arguments associated with each event, such as participants, locations, and temporal attributes (Doddington et al., 2004). Accurate event extraction underpins downstream applications ranging from knowledge graph construction and intelligence analysis to biomedical literature mining, making it a longstanding focus of NLP research (Wang et al., 2020; Li et al., 2021), including continual settings where new event types emerge over time (Le et al., 2024; Dao et al., 2024; Hai et al., 2026).

Recent advances have recast event extraction as a generative task in which a language model produces structured outputs conditioned on the input text, offering a unified paradigm for joint trigger identification, classification, and argument extraction (Du and Cardie, 2020; Lu et al., 2021; Li et al., 2021). Although large language models (LLMs) achieve strong performance within this paradigm, their substantial computational cost renders direct deployment impractical in many real-world settings. Knowledge distillation (KD) provides a principled solution: a frozen LLM teacher guides the training of a compact student model, transferring the teacher’s rich event representations into an efficient architecture suitable for deployment.

Existing KD methods for LLMs, however, are not tailored to the demands of event extraction. Standard KL divergence minimization (Hinton et al., 2015) is susceptible to mode averaging and gradient instability (Wen et al., 2023). Recent methods address these shortcomings through alternative divergences (Wen et al., 2023; Ko et al., 2024; Kim et al., 2026), on-policy and contrastive training (Agarwal et al., 2024; Ko et al., 2025), and structural relationship transfer (Park et al., 2019; Le et al., 2026). Despite these advances, all such methods operate at the token level and treat each output position independently, failing to (1) concentrate the distillation signal on event-relevant tokens such as triggers and arguments, and (2) capture the re-

* Equal contribution.

† Corresponding author: namlh@soict.hust.edu.vn

lational structure among event components that is essential for accurate event understanding.

To address these limitations, we propose **EventKD**, a two-level knowledge distillation framework designed specifically for event extraction. At the token level, we employ a general distributional divergence loss $\mathcal{L}_{KD}$ with event-aware span importance weighting, which concentrates the distillation signal on tokens corresponding to triggers, arguments, and event types. At the span level, we introduce an Event-Aware (EA) Knowledge Distillation loss that preserves pairwise geometric relationships among event-aware spans across multiple Transformer layers. By combining both objectives, our framework enables a compact student to capture not only the teacher’s output behavior but also the internal structural knowledge critical for event understanding. Our contributions are as follows:

- We propose EventKD, a knowledge distillation framework that transfers structured event knowledge from a large teacher LLM to a compact student model, preserving both token-level output behavior and span-level relational structure.
- We introduce an event-aware knowledge distillation approach comprising two complementary components: (1) a general token-level distillation loss $\mathcal{L}_{KD}$ with span importance weighting that focuses the distillation signal on event-relevant positions, and (2) an Event-Aware distillation loss that preserves pairwise structural relationships among event-aware spans across network layers. The framework is agnostic to the choice of divergence; in our experiments, we instantiate $\mathcal{L}_{KD}$ with the Skew Forward KL (SFKL) divergence.
- We conduct extensive experiments on three benchmarks (ACE05, GENEVA and MAVEN), demonstrating consistent improvements over existing distillation methods; additional Llama-3 family and non-English (Spanish) evaluations further validate that EventKD generalizes across architectures and languages.

2 Related Work

LLM-based Event Extraction. Recent advances have shifted event extraction from discriminative classification to generative formulations (Du and Cardie, 2020; Lu et al., 2021; Li et al., 2021), a

trend amplified by LLMs that leverage in-context learning and instruction following (Srivastava et al., 2025; Liu et al., 2025). Despite their strong performance, LLM-based event extractors suffer from critical limitations: they operate purely at the token level without modeling structural relationships between event components, and their computational cost prohibits deployment in resource-constrained settings. While recent reasoning-enhanced approaches (Srivastava et al., 2025; Liu et al., 2025) improve over baseline LLMs through divergent-convergent reasoning chains, they remain confined to a single model’s output distribution and do not transfer structural event knowledge to efficient architectures.

Knowledge Distillation. Conventional KD approaches focus primarily on minimizing output-level divergence while overlooking structural knowledge in intermediate representations (Hinton et al., 2015). Feature-based methods address this gap by aligning internal hidden states or attention maps between student and teacher (Jiao et al., 2020; Sun et al., 2019; Liu et al., 2022; Truong et al., 2025). Recent LLM distillation research has further explored on-policy training (Agarwal et al., 2024), contrastive objectives (Ko et al., 2025), cross-tokenizer alignment via dual-space projection or optimal transport (Zhang et al., 2024; Vu et al., 2026; Vuong et al., 2026; Le Hai et al., 2026), extensions to preference alignment across tokenizers (Nguyen et al., 2026), efficient sparse logit methods (Anshumann et al., 2025), and discrete score-matching objectives that bypass softmax smoothing (Kim et al., 2026); see Xu et al. (2024) for a comprehensive survey. Nevertheless, most existing approaches adopt a static, uniform alignment strategy at the token level, or weight tokens by task-agnostic criteria such as entropy (Vu et al., 2026) or teacher-derived reward (Le et al., 2025), neglecting which tokens carry event-specific semantics and how event components relate structurally, a particular concern under the wide teacher-student capacity gap typical of LLM distillation (Zhang et al., 2023).

Our work addresses these gaps by distilling structural, span-level event knowledge from a frozen LLM teacher into a compact student model. Most closely related, SRA (Dao et al., 2026) and MTA (Chi et al., 2026) also preserve pairwise geometry among span representations, but define spans by tokenizer-offset matching or syntactic phrases and

target open-ended instruction following. In contrast, EventKD derives spans from the event structure of the teacher’s output (triggers, arguments, and types) and explicitly aligns their pairwise relational geometry between teacher and student. This **span-aware, event-aware, relation-aware** distillation preserves the structural topology of event representations that purely generative approaches discard, while simultaneously reducing inference cost.

3 Preliminaries

3.1 Problem Formulation and LLM-based Event Extraction

Problem Formulation. Given an input sentence $\mathbf{x} = (x_1, x_2, \dots, x_N)$ of N tokens, the goal of event extraction is to identify event triggers, classify them into predefined event types from $\mathcal{E}$, and extract their associated arguments. Each extracted event is represented as $e_k = (t_k, c_k, \{(a_{k,j}, r_{k,j})\}_{j=1}^{J_k})$, where t_k is the event trigger (the word or phrase that most clearly evokes the event), $c_k \in \mathcal{E}$ is the event type, and $\{(a_{k,j}, r_{k,j})\}$ denotes the associated arguments with $a_{k,j}$ being an argument span and $r_{k,j} \in \mathcal{R}$ its semantic role (e.g., Agent, Place, Time) (Ebner et al., 2020). Given the full set of extracted events $\mathcal{Y} = \{e_1, e_2, \dots, e_M\}$ (with $\mathcal{Y} = \emptyset$ when no event is present), the structured event representation naturally motivates a span-level approach to knowledge distillation, as we describe in §4.1.

LLM-based Event Extraction. We employ a language model as the Teacher, which performs event extraction using structured prompts. The Teacher is fine-tuned on each dataset using parameter-efficient adaptation (see §5.1 for details). Specifically, given an input sentence $\mathbf{x}$, we construct a prompt $P(\mathbf{x})$ that includes a task instruction describing the event extraction schema, the set of target event types $\mathcal{E}$ with their associated argument roles $\mathcal{R}$, and the input sentence. The Teacher LLM autoregressively generates a structured output sequence $\mathbf{y} = (y_1, y_2, \dots, y_T)$ that encodes the extracted events in a linearized format:

$$\mathbf{y} = f_{\text{LLM}}(P(\mathbf{x}); \theta_T), \quad (1)$$

where θ_T denotes the Teacher parameters and $\mathbf{y}$ is decoded into the structured event set $\mathcal{Y}$. The output sequence $\mathbf{y}$ naturally contains identifiable spans corresponding to event components: trigger

spans, argument spans, and event type regions that encompass complete event annotations. Figure 1 illustrates the prompt design. These spans serve as the foundation for our Event-Aware distillation approach described in §4.1. The model is trained by minimizing the standard autoregressive cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = - \sum_{t=1}^T \log p_{\theta}(y_t \mid \mathbf{y}_{<t}, P(\mathbf{x})), \quad (2)$$

where θ denotes the model parameters and $p_{\theta}(y_t \mid \mathbf{y}_{<t}, P(\mathbf{x}))$ is the predicted probability of the ground-truth token y_t given the preceding tokens and the prompt. This objective applies to both teacher fine-tuning ($\theta = \theta_T$) and student training ($\theta = \theta_S$).

3.2 Knowledge Distillation

Knowledge Distillation (KD) (Hinton et al., 2015) transfers knowledge from a large teacher model to a compact student model by minimizing the divergence between their output distributions. For autoregressive language models, the sequence-level divergence decomposes as a sum over token-level conditionals. Specifically, for a training dataset $\mathcal{D} = \{(\mathbf{x}, \mathbf{y})\}$, the standard forward KL divergence loss is:

$$D_{\text{KL}}(p_T \parallel p_S) \approx \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}} \sum_{t=1}^{|\mathbf{y}|} \sum_{v \in \mathcal{V}} p_{T,t}^v \log \frac{p_{T,t}^v}{p_{S,t}^v}, \quad (3)$$

where $p_{T,t}^v = p_T(v \mid \mathbf{y}_{<t}, \mathbf{x})$ and $p_{S,t}^v = p_S(v \mid \mathbf{y}_{<t}, \mathbf{x})$ denote the teacher and student conditional distributions over the vocabulary $\mathcal{V}$ at time step t , respectively. This formulation enables the student to approximate the teacher’s distribution and capture inter-class correlations (Hinton et al., 2015).

The choice of divergence function significantly affects distillation quality. Forward KL encourages the student to cover all modes of the teacher distribution but suffers from gradient instability when the student assigns near-zero probability to teacher-favored tokens. Reverse KL (RKL) (Wen et al., 2023) instead minimizes $D_{\text{KL}}(p_S \parallel p_T)$, producing mode-seeking behavior that yields sharper student distributions but risks ignoring parts of the teacher’s support. Skew Forward KL (SFKL) (Ko et al., 2024) interpolates a mixture distribution $p_M = \lambda \cdot p_T + (1 - \lambda) \cdot p_S$ into the forward KL target, yielding provably more stable gradients while retaining the mode-covering property. More

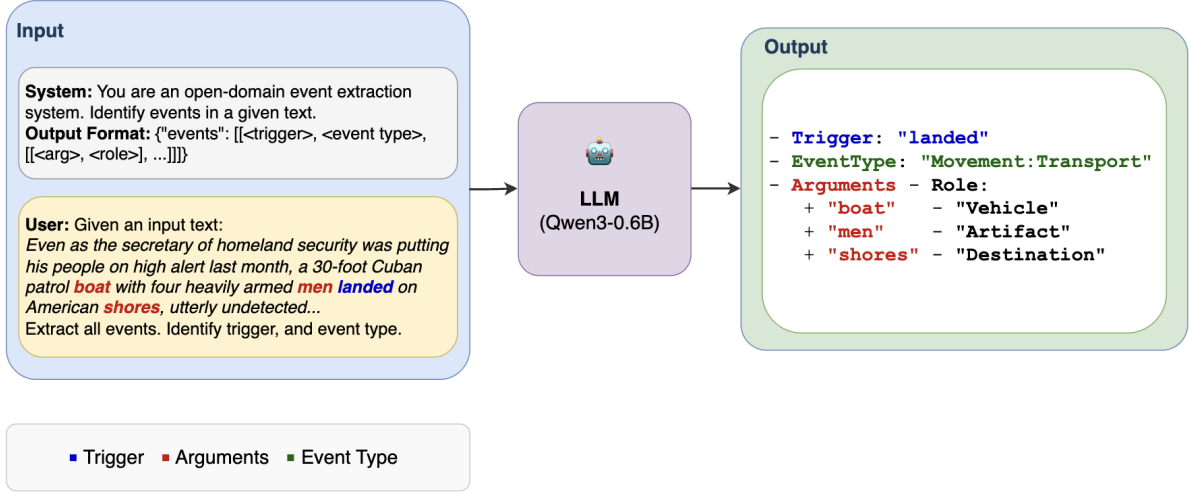


Figure 1: Prompt design for standard event extraction. The model receives the input text and must identify all event triggers and classify their event types.

recently, Concrete Score Distillation (CSD) (Kim et al., 2026) reformulates the distillation objective through a score-matching framework over continuous relaxations of discrete tokens, bypassing the pitfalls of direct distribution matching. We evaluate all three alternatives alongside standard forward KL in our experiments (§5).

4 Event-Aware Knowledge Distillation

This section presents our knowledge distillation framework for LLM-based event extraction. Building on the task formulation and Teacher LLM described in §3.1, we introduce Event-Aware Knowledge Distillation, which transfers structured event knowledge from teacher to student through span-level relational alignment (§4.1), and present the overall training objective (§4.2). An overview of the full framework is provided in Figure 3 (Appendix).

4.1 Event-Aware Knowledge Distillation Loss

As illustrated in Figure 2, in LLM-based event extraction (Liu et al., 2025; Srivastava et al., 2025), the Teacher LLM implicitly performs *event identification* (localizing trigger and argument spans within the input) and *event classification* (assigning each trigger to an event type from the schema $\mathcal{E}$). These two operations leave distinct footprints in the model’s hidden states: identification concentrates attention on salient token spans, while classification encodes type-discriminative features in higher-layer representations.

Our Event-Aware Knowledge Distillation framework is designed to capture both signals through

two complementary objectives. At the *token level*, a distributional divergence loss $\mathcal{L}_{\text{KD}}$ transfers the teacher’s output distribution, preserving the identification and classification decisions encoded in the predicted token probabilities. At the *span level*, our Event-Aware loss $\mathcal{L}_{\text{EA}}$ preserves the geometric structure among event-aware spans, ensuring that the student internalizes not just individual span representations but the full relational web that characterizes an event instance.

Motivation. Conventional generative knowledge distillation forces the student to mimic the teacher’s output token by token in a left-to-right autoregressive manner. While this captures surface-level distributional knowledge, it overlooks the rich structural information inherent in event extraction. Events possess an intrinsic intra-event graph-like structure: each trigger is associated with a specific event type, and arguments depend on both the trigger and their semantic roles within the same event mention. These relational dependencies form a complex web of associations that a purely sequential distillation objective cannot adequately preserve. We therefore propose an *Event-Aware Knowledge Distillation* framework that explicitly transfers the structural event knowledge from the teacher to the student through span-level relational alignment.

Span Representation. To capture the structured nature of events, we define three categories of event-aware spans within the generated output sequence: (1) *Trigger Spans*, the token positions corresponding to event triggers that express event oc-

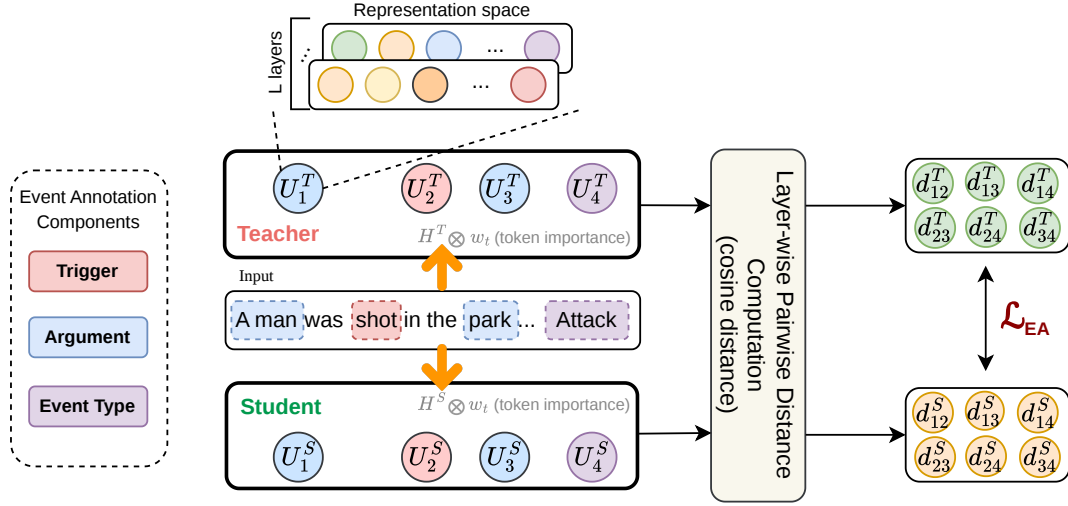


Figure 2: Illustration of Event-Aware Knowledge Distillation. Event-aware spans (triggers, arguments, and event types) are extracted from both teacher and student hidden states. Pairwise cosine distances d_{ij} between spans form a relational structure within each model. The Event-Aware loss $\mathcal{L}_{EA}$ aligns these pairwise distances, preserving the structural geometry of event representations across teacher and student.

currences; (2) *Argument Spans*, the token positions of entities or temporal expressions filling semantic roles in an event (Ebner et al., 2020); and (3) *Event Type Spans*, the token positions of verbalized event type labels in the output.

For each span k at layer l , we compute its representation $U_{k,l}$ as a weighted average of constituent token hidden states:

$$U_{k,l} = \frac{\sum_{t \in S_k} w_{t,l} \cdot H_{t,l}}{\sum_{t \in S_k} w_{t,l}}, \quad (4)$$

where S_k is the set of token indices in span k , $H_{t,l}$ is the hidden state of token t at layer l , and $w_{t,l}$ is a token importance weight. We derive $w_{t,l}$ from a pairwise relevance matrix A_l computed from hidden states by measuring the weighted in-degree of each token (excluding self-loops), followed by standardization across the sequence:

$$w_{t,l} = \frac{\tilde{w}_{t,l}}{\sum_{t'=1}^N \tilde{w}_{t',l}}, \quad \tilde{w}_{t,l} = \sum_{s \neq t} \alpha_{s \rightarrow t,l}, \quad (5)$$

where $\alpha_{s \rightarrow t,l}$ is the pairwise relevance score from token s to token t at layer l (see Appendix A.1 for derivation). Tokens that receive higher attention from other positions are deemed more semantically important within their span.

Event-Aware Loss. The core of our approach is an event-aware loss that preserves the pairwise geometric structure among event spans across

teacher and student. Our formulation adopts the relational span-alignment objective and attention-derived salience weighting of Dao et al. (2026) and Chi et al. (2026), but instantiates them with spans derived from the event structure of the teacher’s output (triggers, arguments, and event types). Rather than matching individual span representations directly, we align the *relational geometry*, that is, how triggers relate to arguments and how different event types are positioned relative to one another in representation space. This relational formulation is inherently more robust to the capacity gap between teacher and student, as it requires the student to reproduce structural relationships rather than absolute representations.

We compute span-level importance weights by aggregating token weights:

$$w_{i,l}^{\text{sp}} = \frac{\tilde{w}_{i,l}^{\text{sp}}}{\sum_{j=1}^{N_l^{\text{sp}}} \tilde{w}_{j,l}^{\text{sp}}}, \quad \tilde{w}_{i,l}^{\text{sp}} = \sum_{t=s_i^l}^{e_i^l} w_{t,l}^{(T)}, \quad (6)$$

where $[s_i^l, e_i^l]$ denotes the token indices of span i , N_l^{sp} is the total number of event-aware spans at layer l , and $w_{t,l}^{(T)}$ is the teacher-derived token importance weight.

Following prior work on intermediate-layer distillation (Romero et al., 2015; Sun et al., 2019; Chi et al., 2026), we select a subset of Transformer layers $L_{\text{key}} \subset \{1, \dots, L\}$, where L is the total number of Transformer layers, from which to extract span

representations (the specific layers are treated as a hyperparameter; see §5 for details). For these key layers, the event-aware loss is defined as:

$$\mathcal{L}_{\text{EA}} = \frac{1}{|L_{\text{key}}|} \sum_{l \in L_{\text{key}}} \sum_{i=1}^{N_l^{\text{sp}}} \sum_{j=i+1}^{N_l^{\text{sp}}} w_{ij, \phi(l)}^{\text{sp}} \cdot \left| d(U_{i, \phi(l)}^T, U_{j, \phi(l)}^T) - d(U_{i, l}^S, U_{j, l}^S) \right|, \quad (7)$$

where $d(\cdot, \cdot)$ denotes cosine distance, $\phi(\cdot)$ denotes the layer mapping function between the student and teacher models, $U_{i, \phi(l)}^T$ and $U_{i, l}^S$ are the teacher and student span representations, respectively, and $w_{ij, l}^{\text{sp}} = w_{i, l}^{\text{sp}} \cdot w_{j, l}^{\text{sp}}$ prioritizes alignment between semantically salient event spans. By minimizing $\mathcal{L}_{\text{EA}}$, the student learns to preserve the within-layer relational structure among event spans, faithfully transferring the geometric relationships that encode the teacher’s event knowledge.

4.2 Training Objective

The complete training objective combines the token-level distillation loss with the event-aware loss:

$$\mathcal{L}_{\text{Total}} = (1 - \beta) \cdot \mathcal{L}_{\text{CE}} + \beta \cdot (\mathcal{L}_{\text{KD}} + \lambda_{\text{EA}} \cdot \mathcal{L}_{\text{EA}}) \quad (8)$$

where $\beta \in [0, 1]$ is the distillation ratio that balances the cross-entropy and distillation objectives, $\mathcal{L}_{\text{CE}}$ is the standard cross-entropy loss defined in Equation 2, $\mathcal{L}_{\text{KD}}$ is the token-level distillation loss between teacher and student output distributions, and λ_{EA} controls the contribution of event-aware alignment. We ablate the choice of $\mathcal{L}_{\text{KD}}$ using Reverse KL (RKL), Skew Forward KL (SFKL), and Concrete Score Distillation (CSD) in §5.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on three widely used event extraction benchmarks: (1) **ACE05** (Dodington et al., 2004), which annotates 33 event types with both trigger and argument annotations across newswire and broadcast documents; (2) **GENEVA** (Parekh et al., 2023), a generative event argument extraction benchmark spanning diverse domains with both trigger and argument annotations; (3) **MAVEN** (Wang et al., 2020), a large-scale trigger detection dataset covering 168 event types from Wikipedia articles. For ACE05 and GENEVA, we report both trigger and argument extraction results; for MAVEN, which focuses on trigger-level evaluation, we report trigger metrics only.

Baselines. We compare against the following representative distillation baselines: (1) **SFT**: the student model fine-tuned on the same training data without any distillation signal; (2) **KD** (Hinton et al., 2015): standard knowledge distillation using forward KL divergence; (3) **RKL** (Wen et al., 2023): distillation using reverse KL divergence, which encourages mode-seeking behavior; (4) **SFKL** (Ko et al., 2024): Skew Forward KL divergence without span-level alignment, isolating the token-level component of our method; (5) **FDD** (Gong et al., 2025): aligning both feature trajectories and their first-order layer-wise derivatives; (6) **DistiLLM** (Ko et al., 2024): a recent method that combines skewed divergences with adaptive off-policy training for LLM distillation; (7) **CSD** (Kim et al., 2026): Concrete Score Distillation, which reformulates knowledge distillation through score matching over continuous relaxations of discrete tokens; and (8) **AMiD** (Shin et al., 2026): α -mixture Assistant Distribution KD, which builds a tunable bridge distribution by mixing teacher and student log-probabilities through an α -divergence family. We also include the **Teacher** model performance as an upper bound reference.

Implementation Details. We use Qwen3-4B-Instruct-2507 (Qwen Team, 2025) as the teacher and Qwen3-0.6B as the student. The teacher is fine-tuned per dataset with LoRA (rank 32, $\alpha = 64$, dropout 0.1); the student is distilled with LoRA (rank 8, $\alpha = 16$, dropout 0.1) for 5 epochs. Both use learning rate 1×10^{-4} with cosine decay, weight decay 1×10^{-2} , per-device batch size 2 with gradient accumulation 8 across 2 GPUs (effective batch size 32), and max sequence length 768. We instantiate $\mathcal{L}_{\text{KD}}$ as the SFKL divergence (Ko et al., 2024) with skew $\lambda = 0.1$, and set $\beta = 0.9$ in Eq. 8. For $\mathcal{L}_{\text{EA}}$, we align two intermediate layers by default (teacher 33, 36 $\rightarrow$ student 25, 28); for ACE05, three layers (30, 33, 36 $\rightarrow$ 22, 25, 28), as the richer schema benefits from deeper alignment. The span loss weight is $\lambda_{\text{EA}} = 2.0$. All experiments use DeepSpeed with bf16 mixed precision on two NVIDIA A100 40GB GPUs.

5.2 Event Extraction Results

Tables 1 and 2 present event extraction results. Table 1 reports event detection performance (trigger identification and type classification) across all three benchmarks. Because argument annotations are available only for ACE05 and GENEVA,

Table 1: Event detection results (Precision, Recall, F1) on three benchmarks. **Bold** indicates the best student result.

Method	ACE05			MAVEN			GENEVA		
	P	R	F1	P	R	F1	P	R	F1
Teacher	77.30	67.93	72.31	63.04	61.96	62.49	62.14	64.83	63.46
SFT	61.84	59.34	60.57	57.20	57.01	57.11	43.82	42.36	43.08
KD	68.02	59.09	63.24	61.72	58.06	59.84	54.43	47.81	50.90
RKL	67.23	60.61	63.75	65.05	57.16	60.85	51.85	46.75	49.17
SFKL	69.45	60.86	64.87	63.58	58.34	60.85	54.77	50.71	52.66
FDD	66.76	58.33	62.26	62.59	57.76	60.08	56.01	48.28	51.86
DistiLLM	66.86	57.58	61.87	63.97	58.31	61.01	54.55	50.45	52.42
CSD	70.62	60.10	64.94	65.21	57.11	60.89	56.86	49.76	53.07
AMiD	68.84	58.59	63.30	61.99	58.45	60.17	53.12	48.23	50.55
EventKD	71.97	62.88	67.12	65.34	59.29	62.17	56.85	53.78	55.27

Table 2: Argument extraction results (Precision, Recall, F1) on ACE05 and GENEVA. **Bold** indicates the best student result.

Method	ACE05			GENEVA		
	P	R	F1	P	R	F1
Teacher	47.87	45.12	46.45	44.26	46.96	45.57
SFT	31.15	31.53	31.34	23.68	23.45	23.57
KD	34.76	33.97	34.36	34.66	30.88	32.66
RKL	34.77	33.62	34.19	30.95	29.85	30.39
SFKL	39.23	37.11	38.14	34.53	32.71	33.60
FDD	31.55	29.79	30.65	31.48	27.34	29.26
DistiLLM	36.35	34.32	35.30	34.61	32.90	33.73
CSD	38.74	35.37	36.98	36.16	33.07	34.54
AMiD	35.56	33.45	34.47	32.21	30.56	31.36
EventKD	45.05	41.99	43.46	38.02	36.19	37.08

Table 3: Llama family generalization with Meta-Llama-3-8B-Instruct (teacher) and Llama-3.2-1B-Instruct (student). All values are F1. MAVEN provides trigger-level annotations only.

Method	Trigger F1			Argument F1	
	ACE05	MAVEN	GENEVA	ACE05	GENEVA
Teacher (Meta-Llama-3-8B-Instruct)	77.23	63.35	70.60	52.46	47.27
SFT	66.30	61.10	52.64	38.00	31.29
DistiLLM	70.69	62.79	64.92	46.03	41.60
CSD	71.25	62.89	65.39	45.32	41.94
AMiD	72.42	62.41	65.14	45.95	41.28
EventKD	72.75	63.05	66.30	46.84	42.74

Table 2 reports argument extraction performance (argument identification and role classification) on these two datasets.

Overall Performance. EventKD consistently achieves the highest F1 scores among all student models across every dataset and metric. On ACE05, EventKD reaches 67.12% trigger F1 and 43.46% argument F1, outperforming the second-best method CSD (64.94% trigger F1) by 2.18 points and SFKL (38.14% argument F1) by 5.32 points. Compared to the SFT lower bound, EventKD improves trigger F1 by 6.55 points and argument F1 by 12.12 points, demonstrating the substantial benefit of distillation. Relative to the Teacher upper

bound (72.31% / 46.45%), EventKD narrows the gap to 5.19 and 2.99 points on trigger and argument F1, respectively. On GENEVA, EventKD achieves 55.27% trigger F1 and 37.08% argument F1, surpassing all baselines. On MAVEN, EventKD reaches 62.17% trigger F1, approaching the teacher (62.49%) with only a 0.32-point gap. To validate Llama family generalization, we additionally evaluate EventKD with a Meta-Llama-3-8B-Instruct teacher and a Llama-3.2-1B-Instruct student ($\sim 6.5\times$ compression). Table 3 shows that EventKD attains the best student F1 on every dataset and metric: +2.06/+0.81 over DistiLLM SFKL on ACE05, +0.91/+0.80 over the strongest baseline (CSD) on GENEVA, and the highest trigger F1 on MAVEN (63.05), confirming that the gains generalize beyond the Qwen family.

Token-Level vs. Span-Level Distillation. To isolate the contribution of $\mathcal{L}_{EA}$, we compare SFKL with and without it. Adding the span-level component improves trigger/argument F1 by +2.25/+5.32 on ACE05 (64.87 $\rightarrow$ 67.12, 38.14 $\rightarrow$ 43.46) and +2.61/+3.48 on GENEVA. The larger argument gain confirms that span-level alignment is particularly beneficial for the role-heavy subtask, where relational dependencies among event components dominate. This asymmetry has a structural explanation: argument extraction requires identifying both the span boundary and its semantic role, which is inherently relational, since the same entity can take different roles depending on which event component it pairs with (e.g., the same entity is the Adjudicator with a regulator trigger but the Entity with the regulated action; see Appendix A.6). Token-level KD constrains only the marginal probability of each output token and cannot enforce such pairwise asymme-

Table 4: Effect of layer mapping on ACE05. Teacher layers are mapped to corresponding student layers for span-level alignment.

Teacher	Student	Trigger F1	Argument F1
[]	[]	64.87	38.14
[36]	[28]	65.59	39.04
[33, 36]	[25, 28]	64.71	40.04
[27, 36]	[19, 28]	67.99	41.55
[28, 32, 36]	[20, 24, 28]	69.21	41.64
[30, 33, 36]	[22, 25, 28]	67.12	43.46
[27, 30, 33, 36]	[19, 22, 25, 28]	66.85	40.25

Table 5: Training cost on ACE05 ($2 \times$ A100 40GB, batch size 16 per device).

Method	Time/step (s)	Avg alloc (GB)	Peak alloc (GB)
RKL	2.67	10.11	22.47
w/ $\mathcal{L}_{EA}$	2.77	10.96	23.32
SFKL	2.53	10.11	22.24
w/ $\mathcal{L}_{EA}$	2.65	10.96	22.62
CSD	2.39	10.11	18.75
w/ $\mathcal{L}_{EA}$	2.56	10.96	19.12

tries, whereas the distances aligned by $\mathcal{L}_{EA}$ preserve this geometric separation in the teacher’s representation space.

5.3 Ablation Study

We conduct ablation experiments on ACE05 to analyze the contribution of each component in our framework. All experiments use the same teacher–student pair (Qwen3-4B / Qwen3-0.6B) and training configuration described in §5.1 unless otherwise noted.

Layer Mapping for Span Alignment. The event-aware loss $\mathcal{L}_{EA}$ aligns span representations extracted from intermediate Transformer layers of the teacher and student. Table 4 examines how the choice of aligned layers affects performance. Aligning no layers (row 1, $\lambda_{EA} = 0$) reduces to token-level distillation only. Using a single last layer provides modest gains, while mapping two evenly spaced layers (e.g., 27, 36 $\rightarrow$ 19, 28) yields stronger improvements, especially for trigger F1. The best overall balance on ACE05 is achieved by three layers (30, 33, 36 $\rightarrow$ 22, 25, 28), which we adopt for this dataset. Adding a fourth layer slightly degrades performance, suggesting that very early layers introduce noise rather than useful structural information.

The effect of varying λ_{EA} is reported in Appendix A.2: increasing λ_{EA} progressively improves both trigger and argument F1, with $\lambda_{EA} = 2$ achieving the best argument F1 (43.46) while maintaining

Table 6: Comparison of token-level KD losses on ACE05, with and without the event-aware loss $\mathcal{L}_{EA}$.

Method	Trigger			Argument		
	P	R	F1	P	R	F1
Teacher	77.30	67.93	72.31	47.87	45.12	46.45
SFT	61.84	59.34	60.57	31.15	31.53	31.34
RKL	67.23	60.61	63.75	34.77	33.62	34.19
w/ $\mathcal{L}_{EA}$	71.14	62.88	66.76	42.48	38.85	40.58
SFKL	69.45	60.86	64.87	39.23	37.11	38.14
w/ $\mathcal{L}_{EA}$	71.97	62.88	67.12	45.05	41.99	43.46
CSD	70.62	60.10	64.94	38.74	35.37	36.98
w/ $\mathcal{L}_{EA}$	72.19	64.90	68.35	40.91	39.20	40.04

competitive trigger F1 (67.12).

Choice of Token-Level Distillation Loss. Table 6 compares three token-level divergence functions – Reverse KL (RKL), Skew Forward KL (SFKL), and Concrete Score Distillation (CSD) – both with and without $\mathcal{L}_{EA}$. Without $\mathcal{L}_{EA}$, RKL yields the lowest F1, as its mode-seeking behavior causes the student to focus on a narrow subset of the teacher’s distribution, missing less frequent event types and roles. CSD achieves the highest trigger F1 (64.94) through smoother score-matching gradients, but lags on arguments (36.98 vs. 38.14 for SFKL), suggesting that continuous relaxation is less effective for discrete argument structures. SFKL strikes the best balance by interpolating a mixture distribution that stabilizes gradients while preserving mode-covering behavior. Adding $\mathcal{L}_{EA}$ improves all three divergences, with the largest gains on RKL (+6.39 argument F1), indicating that weaker token-level objectives benefit more from span-level structural guidance. We adopt SFKL as the default because it achieves the best argument F1 (43.46) with $\mathcal{L}_{EA}$, and argument extraction is the harder subtask.

Computational Cost. Table 5 shows the training cost of adding $\mathcal{L}_{EA}$ on ACE05 using 2 A100 40GB GPUs (batch size 16). The overhead is modest: step time increases by <5% (2.53 $\rightarrow$ 2.65 s for SFKL) and memory by $\sim$ 0.85 GB, with no change in peak allocation tier. Beyond training, Table 7 reports the inference-time comparison between teacher and student: EventKD retains 92.8% / 93.6% of teacher Trigger / Argument F1 (see Tables 1–2) while delivering $6.7\times$ parameter compression, $1.74\times$ higher throughput, and $4.3\times$ lower peak memory; all KD variants share this footprint since they reuse the same Qwen3-0.6B backbone.

Table 7: Teacher vs. student inference cost on ACE05 (single A40, bf16, batch size 16, fixed-length generation of 128 tokens; averaged over 3 runs).

Model	Params	Tokens/s	Peak Mem (GB)
Teacher (Qwen3-4B + LoRA SFT)	4.02B	151.9	9.25
Student EventKD (ours)	0.60B	264.6	2.17
Ratio (Student / Teacher)	0.15×	1.74×↑	0.23×↓

6 Conclusion

We presented EventKD, a two-level knowledge distillation framework for event extraction. At the token level, a distributional divergence loss $\mathcal{L}_{KD}$ (instantiated with SFKL) aligns teacher and student output distributions. At the span level, an Event-Aware loss $\mathcal{L}_{EA}$ preserves pairwise geometric relationships among event-aware spans (triggers, arguments, event types) across Transformer layers. Combining both objectives lets a compact student capture the teacher’s output behavior together with its internal structural knowledge, yielding consistent improvements over existing distillation methods on ACE05, GENEVA, and MAVEN.

7 Limitations

Our framework shares several limitations common to LLM-based knowledge distillation approaches. First, the student’s performance is upper-bounded by the teacher’s competence on the target task. Second, our experiments primarily focus on English-language benchmarks; preliminary non-English results on MINION Spanish (Appendix A.5) suggest the framework transfers cleanly to other languages, but a thorough evaluation across morphologically rich and low-resource languages remains an open question. Third, we do not explore interactions with complementary compression techniques such as quantization or pruning, which may offer additional efficiency gains orthogonal to representation-level distillation. Finally, due to computational resource constraints, our empirical study is limited to two LLM families (Qwen3 and Llama-3) and the event-extraction domains covered by ACE05, GENEVA, and MAVEN; broader exploration across additional model architectures and downstream domains is left to future work.

Acknowledgments

Trung Le was supported by the Air Force Office of Scientific Research under award number FA2386-25-1-4023 and the ARC Discovery Project grant DP250100262. Thien Huu Nguyen was partially

supported by NSF Grant #2239570. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. Nam Le Hai was funded by the PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), VinUniversity, code VINIF.2025.TS68.

References

- Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. [On-policy distillation of language models: Learning from self-generated mistakes](#). In *The Twelfth International Conference on Learning Representations (ICLR)*.
- Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. 2025. [Sparse logit sampling: Accelerating knowledge distillation in LLMs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18085–18108, Vienna, Austria. Association for Computational Linguistics.
- Pham Khanh Chi, Quoc Phong Dao, Thuat Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026. [MTA: Multi-granular trajectory alignment for large language model distillation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32669–32684, San Diego, California, United States. Association for Computational Linguistics.
- Quoc Phong Dao, Hoang Son Nguyen, Pham Khanh Chi, Tung Nguyen, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Trung Le. 2026. [SRA: Span representation alignment for large language model distillation](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32961–32975, San Diego, California, United States. Association for Computational Linguistics.
- Viet Dao, Van-Cuong Pham, Quyen Tran, Thanh-Thien Le, Linh Van Ngo, and Thien Huu Nguyen. 2024. [Lifelong event detection via optimal transport](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12610–12621, Miami, Florida, USA. Association for Computational Linguistics.
- George R. Doddington, Alexis Mitchell, Mark A. Przybocki, Lance A. Ramshaw, Stephanie M. Strassel, and Ralph M. 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)*, pages 837–840, Lisbon, Portugal. European Language Resources Association (ELRA).

- Xinya Du and Claire Cardie. 2020. [Event extraction by answering \(almost\) natural questions](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.
- Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. [Multi-sentence argument linking](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.
- Guoqiang Gong, Jiaying Wang, Jin Xu, Deping Xiang, Zicheng Zhang, Leqi Shen, Yifeng Zhang, Junhua Shu, Zhaolong Xing, Zhen Chen, Pengzhang Liu, and Ke Zhang. 2025. [Beyond logits: Aligning feature dynamics for effective knowledge distillation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23067–23077, Vienna, Austria. Association for Computational Linguistics.
- Nam Le Hai, Linh Ngo Van, and Sang Dinh. 2026. [Mozilla: Continual event detection through the lens of multi-objective optimization and language model head preservation](#). *Computational Linguistics*, pages 1–46.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. [Distilling the knowledge in a neural network](#). *arXiv preprint arXiv:1503.02531*. NIPS 2014 Deep Learning and Representation Learning Workshop.
- Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. [TinyBERT: Distilling BERT for natural language understanding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.
- Yeongmin Kim, Donghyeok Shin, Mina Kang, Byeonghu Na, and Il-chul Moon. 2026. [Distillation of large language models via concrete score matching](#). In *The Fourteenth International Conference on Learning Representations (ICLR)*.
- Jongwoo Ko, Tianyi Chen, Sungnyun Kim, Tianyu Ding, Luming Liang, Ilya Zharkov, and Se-Young Yun. 2025. [DistiLLM-2: A contrastive approach boosts the distillation of LLMs](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pages 31044–31062. PMLR.
- Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. 2024. [DistiLLM: Towards streamlined distillation for large language models](#). In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pages 24872–24895. PMLR.
- Thanh-Thien Le, Viet Dao, Linh Van Nguyen, Thi-Nhung Nguyen, Linh Ngo Van, and Thien Huu Nguyen. 2024. [SharpSeq: Empowering continual event detection through sharpness-aware sequential-task learning](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3632–3644, Mexico City, Mexico. Association for Computational Linguistics.
- Tue Le, Hoang Tran, Quyen Tran, Linh Ngo, Mehrtash Harandi, and Trung Le. 2025. [Token-level self-play with importance-aware guidance for large language models](#). *Advances in Neural Information Processing Systems*, 38:120829–120864.
- Vinh Le, Nguyen Hong Dang, Tu Vu, Linh Ngo Van, Duc Anh Nguyen, and Trung Le. 2026. [Hierd: Hierarchical relational distillation for vision-language embedding models](#). In *Forty-third International Conference on Machine Learning*.
- Nam Le Hai, Anh Duc Le, Tu Vu, and Linh Ngo Van. 2026. [Discord: Enhancing knowledge distillation via cross-chain of thought and optimal transport alignment between models with different tokenizers](#). *Knowledge-Based Systems*, page 116606.
- Sha Li, Heng Ji, and Jiawei Han. 2021. [Document-level event argument extraction by conditional generation](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.
- Chang Liu, Chongyang Tao, Jiazhan Feng, and Dongyan Zhao. 2022. [Multi-granularity structural knowledge distillation for language model compression](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1001–1011, Dublin, Ireland. Association for Computational Linguistics.
- Wenxuan Liu, Zixuan Li, Long Bai, Yuxin Zuo, Daozhu Xu, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. [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*, pages 34377–34399, Suzhou, China. Association for Computational Linguistics.
- Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. [Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction](#). 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)*, pages 2795–2806, Online. Association for Computational Linguistics.
- Truong Nguyen, Phi Van Dat, Ngan Nguyen, Linh Ngo Van, Trung Le, and Thanh Hong Nguyen. 2026.

- CTPD: cross tokenizer preference distillation. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI, pages 37783–37790. AAAI Press.
- Tanmay Parekh, I-Hung Hsu, Kuan-Hao Huang, Kai-Wei Chang, and Nanyun Peng. 2023. **GENEVA: Benchmarking generalizability for event argument extraction with hundreds of event types and argument roles**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3664–3686, Toronto, Canada. Association for Computational Linguistics.
- Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. **Relational knowledge distillation**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3962–3971.
- Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022. **MINION: a large-scale and diverse dataset for multilingual event detection**. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2286–2299, Seattle, United States. Association for Computational Linguistics.
- Qwen Team. 2025. **Qwen3 technical report**. *arXiv preprint arXiv:2505.09388*.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. **FitNets: Hints for thin deep nets**. In *3rd International Conference on Learning Representations (ICLR)*.
- Donghyeok Shin, Yeongmin Kim, Suhyeon Jo, Byeonghu Na, and Il-chul Moon. 2026. **AMiD: Knowledge distillation for LLMs with α -mixture assistant distribution**. In *The Fourteenth International Conference on Learning Representations (ICLR)*.
- Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. **Instruction-tuning LLMs for event extraction with annotation guidelines**. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13055–13071, Vienna, Austria. Association for Computational Linguistics.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. **Patient knowledge distillation for BERT model compression**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4322–4331, Hong Kong, China. Association for Computational Linguistics.
- Minh-Phuc Truong, Hai An Vu, Tu Vu, Nguyen Thi Ngoc Diep, Linh Ngo Van, Thien Huu Nguyen, and Trung Le. 2025. **EMO: Embedding model distillation via intra-model relation and optimal transport alignments**. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7594–7606, Suzhou, China. Association for Computational Linguistics.
- Duc Trung Vu, Pham Khanh Chi, Dat Phi Van, Linh Ngo Van, Dinh Viet Sang, and Trung Le. 2026. **DWA-KD: Dual-space weighting and time-warped alignment for cross-tokenizer knowledge distillation**. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3513–3527, Rabat, Morocco. Association for Computational Linguistics.
- Hoang Tran Vuong, Tue Le, Quyen Tran, Linh Ngo Van, and Trung Le. 2026. **MCW-KD: Multi-cost Wasserstein knowledge distillation for large language models**. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, pages 33332–33340. AAAI Press.
- Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. **MAVEN: A massive general domain event detection dataset**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.
- Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. **f-divergence minimization for sequence-level knowledge distillation**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10817–10834, Toronto, Canada. Association for Computational Linguistics.
- Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024. **A survey on knowledge distillation of large language models**. *arXiv preprint arXiv:2402.13116*.
- Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023. **Lifting the curse of capacity gap in distilling language models**. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4535–4553, Toronto, Canada. Association for Computational Linguistics.
- Songming Zhang, Xue Zhang, Zengkui Sun, Yufeng Chen, and Jinan Xu. 2024. **Dual-space knowledge distillation for large language models**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18164–18181, Miami, Florida, USA. Association for Computational Linguistics.

A Appendix

A.1 Token Importance Computation

This section provides a detailed derivation of the token importance weights $w_{t,l}$ used in the span

representation (Eq. 4). Rather than treating all tokens within a span equally, we assign each token a scalar importance score that reflects how much contextual relevance it carries, as determined by the collective attention it receives from other positions in the sequence. This mechanism ensures that semantically salient tokens, such as trigger words and argument heads, contribute more prominently to the span-level representation.

Hidden State Normalization. Given the hidden state $H_{t,l} \in \mathbb{R}^d$ of token t at Transformer layer l , we first apply variance-based normalization to obtain a standardized representation:

$$\hat{H}_{t,l} = \frac{H_{t,l}}{\sigma(H_{t,l})}, \quad (9)$$

where $\sigma(H_{t,l})$ denotes the standard deviation computed across the d dimensions of $H_{t,l}$. This normalization step removes magnitude differences across tokens and layers, allowing the subsequent attention computation to focus purely on directional similarity between token representations.

Pairwise Attention Scoring. We compute a pairwise relevance score between every pair of tokens (s, t) in the sequence using scaled dot-product attention on the normalized hidden states. To prevent trivial self-reinforcement and to handle variable-length sequences, we apply a masking function that excludes self-loops and padding positions:

$$S_{s \rightarrow t,l} = \frac{\hat{H}_{s,l}^\top \hat{H}_{t,l}}{\sqrt{d}} + M_{s,t}, \quad (10)$$

where the mask $M_{s,t}$ is defined as:

$$M_{s,t} = \begin{cases} -\infty & \text{if } s = t \text{ or } t \in \mathcal{P}, \\ 0 & \text{otherwise,} \end{cases} \quad (11)$$

and $\mathcal{P}$ is the set of padding token indices. Setting $M_{s,s} = -\infty$ eliminates self-attention, ensuring that a token’s importance is determined solely by how other tokens attend to it. Masking out padding positions prevents them from influencing the importance scores.

Attention Weight Computation. The raw scores are converted into proper probability distributions via a row-wise softmax over valid (non-masked) positions:

$$\alpha_{s \rightarrow t,l} = \frac{\exp(S_{s \rightarrow t,l})}{\sum_{u=1}^N \exp(S_{s \rightarrow u,l})}, \quad (12)$$

where N is the total number of non-padding tokens. The coefficient $\alpha_{s \rightarrow t,l}$ quantifies the degree to which token s attends to token t at layer l .

Importance Aggregation. Finally, the importance weight of token t at layer l is obtained by averaging the attention it receives from all other tokens in the sequence:

$$w_{t,l} = \frac{1}{N} \sum_{s=1}^N \alpha_{s \rightarrow t,l}. \quad (13)$$

Intuitively, $w_{t,l}$ measures the mean inbound attention directed toward token t : tokens that are consistently attended to by many other positions receive higher importance scores. This attention-based aggregation naturally assigns greater weight to tokens that serve as central anchors in the contextual representation, such as event triggers and salient argument tokens, making it well-suited for event-aware span pooling in Eq. 4.

A.2 Effect of λ_{EA}

Table 8 shows the effect of varying the event-aware loss weight λ_{EA} in the total objective. When $\lambda_{EA} = 0$, the model reduces to pure token-level distillation. Increasing λ_{EA} progressively improves both trigger and argument F1, with $\lambda_{EA} = 2$ achieving the best argument F1 (43.46) while maintaining competitive trigger F1 (67.12). Beyond this point ($\lambda_{EA} = 3$), argument performance drops noticeably, indicating that an overly strong span-level signal can interfere with token-level learning.

Table 8: Effect of λ_{EA} on ACE05.

λ_{EA}	Trigger			Argument		
	P	R	F1	P	R	F1
0	69.45	60.86	64.87	39.23	37.11	38.14
0.5	73.28	64.39	68.55	42.31	38.85	40.51
1	72.38	62.88	67.30	44.46	41.99	43.19
2	71.97	62.88	67.12	45.05	41.99	43.46
3	72.22	62.37	66.94	40.95	39.02	39.96

A.3 Distillation Weight Sensitivity (β)

To examine the sensitivity of EventKD to the distillation ratio β in the total objective (Eq. 8, where $\mathcal{L}_{\text{Total}} = (1 - \beta) \cdot \mathcal{L}_{\text{CE}} + \beta \cdot (\mathcal{L}_{\text{KD}} + \lambda_{EA} \cdot \mathcal{L}_{EA})$), we sweep $\beta \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ on ACE05 using the same Qwen3-0.6B / Qwen3-4B setup, reporting best F1 across 5 epochs.

Performance is stable across a broad range ($\beta \in [0.3, 0.9]$) with no sharp degradation. The choice

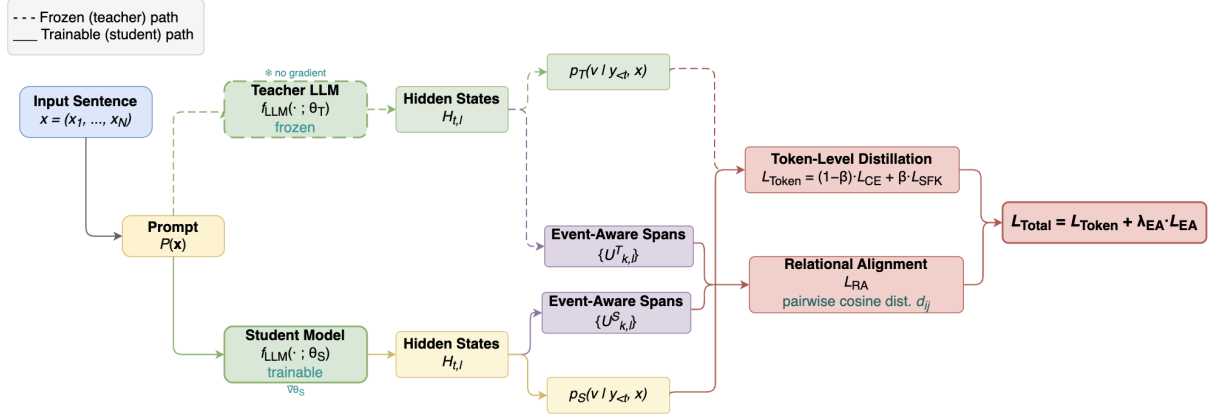


Figure 3: Overview of our knowledge distillation framework for LLM-based event extraction. The input sentence is processed through a prompt template and fed to both the frozen Teacher LLM and the trainable Student model. Token-level distillation ($\mathcal{L}_{\text{KD}}$) aligns output distributions using Skew Forward KL divergence. Structural distillation ($\mathcal{L}_{\text{EA}}$) extracts event-aware spans from hidden states and aligns their pairwise relational geometry between teacher and student.

Table 9: Sensitivity of EventKD to the distillation ratio β on ACE05.

β	0.1	0.3	0.5	0.7	0.9 (paper)	1.0
Trigger F1	66.7	67.5	66.7	68.8	67.1	60.5
Argument F1	40.8	41.8	42.2	40.9	43.5	30.0

$\beta = 0.9$ yields the best Argument F1 while maintaining competitive Trigger F1. Importantly, performance drops significantly at $\beta = 1.0$ (i.e., when the cross-entropy anchor is removed), e.g., a -13.5 Argument F1 drop relative to $\beta = 0.9$, confirming that while distillation should be emphasized, the supervised cross-entropy signal remains essential.

A.4 Uniform vs. Event-Aware Pairwise Alignment

Prior structural / multi-granularity KD approaches align hidden states or pairwise interactions across all tokens, either uniformly (e.g., Jiao et al., 2020; Liu et al., 2022) or weighted by task-agnostic saliency (Chi et al., 2026), without regard to task-specific semantics. To isolate the benefit of EventKD’s task-aware span selection (vs. task-agnostic alignment), we train a controlled *Token-level Pairwise KD* baseline that shares EventKD’s setup exactly (Qwen3-0.6B/4B, ACE05, identical layer mapping, $\beta = 0.9$, $\lambda_{\text{EA}} = 2.0$) but replaces event-aware span selection with uniform token-pair alignment over all positions.

EventKD outperforms the uniform Token-level Pairwise KD baseline on both metrics, by $+0.72$ Trigger F1 and $+3.79$ Argument F1. The

Table 10: Event-aware vs. uniform pairwise alignment on ACE05.

Method	Trigger F1	Argument F1
Token-level Pairwise KD (uniform)	66.40	39.67
EventKD (event-aware, ours)	67.12	43.46

gap widens sharply on the role-heavy argument-extraction subtask where structural relations between event components are decisive: under a wide teacher–student capacity gap, focusing the student’s limited capacity on event-aware structure (rather than uniform token-pair alignment) yields larger gains precisely on the metric that requires distinguishing roles.

A.5 Non-English Generalization on MINION Spanish

The proposed framework is language-agnostic by construction: span extraction operates on character offsets within the teacher’s structured JSON output and does not depend on language-specific tokenization, morphology, or word segmentation. To validate this empirically, we run the Qwen3 pipeline on the Spanish split of MINION (Pouan Ben Veyseh et al., 2022), a multilingual event-detection benchmark covering eight typologically diverse languages. Both teacher and student are fine-tuned on the Spanish MINION training split using the same protocol as our Qwen3 setup in §5.1 (5 epochs, LoRA-based PEFT, $\beta = 0.9$, $\lambda_{\text{EA}} = 2.0$, default 2-layer mapping). MINION provides only trigger-level annotations, so we report trigger F1 only.

Table 11: Non-English generalization on MINION Spanish (trigger-only).

Method	Trigger F1
Teacher (Qwen3-4B + LoRA)	62.72
Student SFT	58.30
DistiLLM (SFKL)	58.47
CSD	59.15
EventKD (ours)	60.20

EventKD outperforms the SFT baseline by +1.90 Trigger F1, DistiLLM (SFKL) by +1.73, and CSD by +1.05, while retaining 96.0% of teacher trigger F1 under $6.7\times$ parameter compression. The smaller absolute gain over CSD on MINION (+1.05) compared to ACE05 (+2.18) is consistent with MINION being a trigger-only dataset: without argument spans, EventKD’s event-aware span set reduces to triggers and event-type tokens only, narrowing the structural headroom that span-level alignment can exploit. EventKD nonetheless transfers cleanly to a non-English setting and outperforms every KD baseline tested.

A.6 Qualitative Case Studies on ACE05

To complement the aggregate scores in Tables 1–2, we present one representative success case and one shared failure case drawn from the ACE05 test split, contrasting EventKD with the two strongest KD baselines, DistiLLM (SFKL) and CSD, under the Llama-3 (8B→1B) family setting.

Success case (role disambiguation and multi-trigger recovery). *Input A (idx 108):* “Last month, the SEC slapped fines totaling 1.4 billion dollars on 10 Wall Street brokerages to settle charges of conflicts of interest between analysts and investors.”

Method	Trigger / Type	Arguments
Gold	fines / Justice:Fine	(SEC, Adjudicator), (brokerages, Entity)
EventKD (ours)	fines / Justice:Fine	(SEC, Adjudicator) ✓, (brokerages, Entity) ✓
DistiLLM (SFKL)	fines / Justice:Fine	(SEC, <i>Enforcer</i> ×), (brokerages, Entity) ✓
CSD	fines / Justice:Fine	(SEC, <i>Entity</i> ×), (brokerages, Entity) ✓

Input B (idx 37): “Anwar, 56, who this week completed four years in prison on a corruption charge, now faces an earliest possible release date of April 14, 2009 if he is given one third remission of his sentence for good behaviour.”

In Input A, all three methods recover the trigger (fines) and event type (Justice:Fine), but only EventKD assigns the gold Adjudicator role to

Method	Trigger / Type	Arguments
Gold	charge / release / sentence (3 events)	(Anwar, Person) on ‘release’
EventKD (ours)	release, sentence (2/3)	—
DistiLLM (SFKL)	release (1/3)	(Anwar, Person) ✓
CSD	released × (0/3)	(Anwar, Person) ✓

SEC; DistiLLM mislabels it as *Enforcer* and CSD as *Entity*, collapsing the regulator and the regulated entity into the same role. In Input B, which packs three closely related Justice events into one sentence, EventKD recovers two of three triggers with exact gold spans, DistiLLM only one, and CSD zero. Both phenomena are consistent with the design of $\mathcal{L}_{EA}$: distilling pairwise distances between the trigger span and each argument / co-occurring event span preserves the asymmetry between regulator and regulated and the relative geometry between sibling triggers, neither of which token-level KD enforces. The trade-off is visible in Input B: EventKD drops the (Anwar, Person) argument that the baselines retain, so multi-trigger recall comes at a small per-event argument-recall cost.

Shared failure case. *Input (idx 158):* “She is being held on 50,000 dollars bail on a charge of first-degree reckless homicide and hiding a corpse in the death of the infant born in January.” This sentence contains 5 gold triggers (held, charge, homicide, death, born); all three methods recover only the same 2 (charge, death), missing held, homicide (a second Life:Die in the same sentence), and born. The shared failure mode (sentences with many lexically distinct triggers in close proximity and the same broad event class repeating) points to a limitation that span-level alignment alone cannot resolve when the teacher itself does not produce a sufficiently disambiguating signal across nearly co-located spans; addressing it would likely require explicit coreference or position-aware objectives, which we mark as future work.