

Article

Structure-Aware Generative Information Extraction via Feature Space Alignment

Yuanqing Li, Chen Tao, Baoyu Zhang and Weishan Zhang * 

Shandong Data Open Innovative Application Laboratory, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China; s23070036@s.upc.edu.cn (Y.L.); dongshou2015@s.upc.edu.cn (C.T.); b22070002@s.upc.edu.cn (B.Z.)

* Correspondence: zhangws@upc.edu.cn

Abstract

Large language models (LLMs) face difficulties in leveraging the syntactic structures and entity relations embedded in text for long-document information extraction. To address this issue, this paper proposes a generative extraction method integrating heterogeneous topology awareness and spatial alignment. The method first extracts syntactic and coreference information to construct a heterogeneous document graph and employs a mixture-of-experts network to decouple and encode multi-type topological features. A component orthogonal projection mechanism and a graph-text contrastive learning strategy are then utilized to align the extracted graph features to the underlying semantic space of the language model with high fidelity. Furthermore, Topology-Aware Encoder compresses the global features into fixed-length structural prompts to guide text generation. Experiments on the ACE2005, WikiEvents, and DuEE datasets demonstrated that the proposed method achieved state-of-the-art performance on information extraction tasks. Consequently, these results suggest that the proposed framework is a promising approach for complex information extraction across base LLMs of different scales.

Keywords: multimodal fusion; information extraction; graph neural network; large language model

1. Introduction

Information Extraction (IE) is a foundational task in natural language processing (NLP), which aims to accurately identify and extract structured knowledge—such as entities, relations, and events—from massive unstructured texts. It serves as a critical prerequisite for constructing domain-specific knowledge graphs, enabling intelligent retrieval, and supporting public opinion analysis [1,2]. In recent years, with the evolution of pre-training paradigms, IE technology has undergone profound transformations. Large Language Models (LLMs) represented by GPT and LLaMA have demonstrated remarkable capabilities in semantic understanding and zero-shot generalization [3]. Through techniques like instruction tuning and prompt engineering [4,5], generative LLMs can unify heterogeneous extraction tasks into a sequence-to-sequence generation process, thereby greatly simplifying the cumbersome module cascading inherent in traditional pipeline models.

Despite the significant success of the generative paradigm in short texts and general scenarios, it still faces severe challenges when dealing with long-document information extraction characterized by complex dependencies. First, relationships between entities in long documents often span multiple sentences, involving intricate coreference chains and



Academic Editors: Jennifer D'Souza and Chengzhi Zhang

Received: 9 March 2026

Revised: 18 April 2026

Accepted: 19 April 2026

Published: 24 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

long-distance syntactic dependencies. Existing LLMs are essentially autoregressive models based on one-dimensional sequences and self-attention mechanisms. Constrained by the quadratic complexity of attention computation and the issue of long-sequence forgetting, they struggle to explicitly and globally capture the latent non-Euclidean topological structures within the text [6]. This “structural perception blind spot” makes the model highly susceptible to losing crucial clues when inferring complex argument relationships.

Second, while Graph Neural Networks (GNNs) possess unique advantages in processing topological structures [7,8], directly injecting GNN features into LLMs remains a daunting task. There exists a massive modality gap between the graph feature space extracted by GNNs and the pre-trained textual semantic space of LLMs [9]. Simple feature concatenation often leads to context shift in large models, which not only fail to utilize structural information effectively but also introduce severe structural noise, disrupting the original language generation capabilities of the models [10]. Recent studies on Graph-Enhanced LLMs have explored external graph adapters or prompts to alleviate this issue [11,12], but they often overlook the inherent geometric distribution differences between the two modalities, leading to suboptimal alignment.

Compared with existing graph-enhanced LLM paradigms, our framework follows a non-invasive graph integration strategy. Previous methods typically either linearize graph structure into textual descriptions, which may introduce excessive context burden and aggravate the lost-in-the-middle issue, or inject graph modules deeply into the internal architecture of the LLM, which increases integration complexity and may disturb the original semantic space of the pre-trained model. In contrast, our method completes graph-feature alignment and injection without modifying the internal Transformer backbone, enabling the LLM to exploit structural knowledge through aligned structural prompts. Moreover, instead of directly concatenating variable-length graph tokens, the proposed Topology-Aware Encoder (TAE) employs topology-masked sparse cross-attention to selectively aggregate graph neighborhoods and compress full-graph representations into fixed-length, high-density structural prompts, thereby reducing the adverse effect of overly long contexts on the LLM.

To address the above issues, this paper proposes a generative information extraction framework that integrates heterogeneous topology awareness with cross-modal feature alignment. The contributions of this work are threefold. First, we construct a heterogeneous document graph that combines syntactic dependency, cross-sentence semantic association, and candidate argument connectivity, and we encode this graph with a context-aware multi-expert adapter. Second, we introduce a joint alignment strategy that combines PCA-based projection with contrastive objectives to improve compatibility between graph features and the LLM embedding space. Third, we design a topology-aware encoder that compresses variable-size graph representations into fixed-length structural prompts for downstream generation. Empirically, these components improve extraction performance on three benchmark datasets, including ACE2005, WikiEvents [13], and DuEE [14].

The rest of this paper is organized as follows. Section 2 reviews related work on generative information extraction, graph neural networks for IE, and graph-enhanced large language models. Section 3 presents the proposed methodology, including heterogeneous document graph construction, the context-aware MoE Graph Adapter, cross-modal semantic-structure alignment, and the Topology-Aware Encoder. Section 4 describes the datasets, baselines, evaluation metrics, and implementation details. Section 5 reports the main experimental results, ablation studies, error analysis, efficiency and robustness analysis, and case study. Finally, Section 6 concludes the paper.

2. Related Works

2.1. Generative Information Extraction

Traditional information extraction methods primarily rely on sequence labeling or joint extraction architectures. These approaches typically require designing complex network layers and tag systems tailored to specific tasks, making it difficult to adapt to open-domain pattern transfers [15]. With the rise of large-scale pre-trained models, generative information extraction has gradually become mainstream. The UIE framework proposed by Lu et al. [16] introduced a unified Structural Extraction Language (SEL), unifying entity, relation, and event extraction tasks into a text generation problem for the first time. InstructUIE [4] further introduced instruction tuning techniques, constructing datasets containing multi-task instructions, endowing the model with powerful multi-task joint extraction and zero-shot generalization capabilities.

To further improve extraction accuracy and adaptability, recent studies have explored various prompt engineering and reasoning enhancement techniques. For instance, CodeKGC [17] explores utilizing the structural features of code to transform extraction tasks into code generation tasks that adhere to specific programming specifications, aiming to obtain stronger structural constraints. ChatUIE [18] guides the model to progressively complete extraction through interactive conversational forms. Furthermore, recognizing that complex IE scenarios often require multi-step logical deduction, recent approaches like MR-UIE [19] have emphasized the importance of multi-perspective reasoning. By integrating reinforcement learning (RL) with multi-perspective reasoning paths, MR-UIE transitions LLMs from passive extractors to active reasoners, significantly improving generalization in zero-shot settings. Other approaches have explored prompting argument interactions [20] or framing extraction as a question-answering task [21] to guide the generation process. Although these generative methods have significantly lowered the task threshold and improved reasoning, they still fundamentally treat text as a flat, one-dimensional sequence. This lack of explicit modeling for complex non-Euclidean structures in long documents limits their performance when dealing with cross-sentence or long-distance dependencies.

2.2. Graph Neural Networks for Information Extraction

Deep learning models, with their deep hierarchical architectures, possess excellent nonlinear feature fitting capabilities. In IE tasks, this allows them to more comprehensively mine deep semantic information in texts, overcoming the limitations of traditional methods when handling complex semantic associations [22]. For instance, Yan et al. [23] proposed the MOGANED model, which effectively aggregates multi-hop neighborhood information of candidate triggers through a high-order syntactic graph computation mechanism. Addressing the issue that traditional dependency tree GCNs easily ignore the semantics of edge labels, Cui et al. [24] proposed an edge-enhanced graph convolutional network (EE-GCN), which jointly models syntactic dependency structures and relation labels to more accurately capture word interactions.

To resolve the error propagation issue in pipeline methods and capture long-distance dependencies, Wadden et al. [25] proposed the DyGIE++ framework. This model dynamically constructs a Span Graph to update text representations and propagates information across named entity recognition, relation extraction, and event extraction tasks using shared span representations, significantly improving extraction performance on overlapping events. While GNNs have proven to be the de facto standard for capturing structural priors, most existing GNN-based IE models still rely on specific task heads and isolated training paradigms, making it difficult to integrate them seamlessly with the versatile generative capabilities of foundation models.

2.3. Graph-Enhanced Large Language Models

With the emergence of powerful LLMs, aligning graph data as an independent modality with LLMs—often referred to as Graph4LLM—has developed into a crucial research direction to mitigate LLMs’ inherent limitations such as hallucination and lack of structural reasoning [26,27]. Current research generally integrates graphs into LLMs at different stages of the pipeline.

Input-Phase Integration: A prominent line of work incorporates graph structures directly into the input prompts. Frameworks like GraphPrompt [28] and LLaGA [29] have explored unifying pre-training and downstream tasks or developing graph assistants. Methods like GraphRAG [30] construct corpus-specific graph indices over unstructured texts to enable macro-level understanding and query-specific summarization. Similarly, Prompt from Instance-Level Graph approaches, such as GraphReader [31], dynamically organize long documents into atomic fact graphs to mitigate the “lost-in-the-middle” issue during long-context reasoning. While effective for retrieval, linearizing large graphs into text sequences often leads to excessively long inputs and potential loss of high-order topological information.

Model-Phase Integration: To achieve deeper alignment, researchers attempt to fuse graph modules directly with the LLM architecture. Some studies utilize External Graph Adapters to project graph features into the LLM latent space without altering core parameters. For example, GraphAdapter [11] employs GNNs as efficient adapters for LLMs to tackle text-attributed graphs, training the adapter via auto-regression on node text. Other works explore Internal Model Fusion, where graph layers are interleaved with Transformer layers. A recent notable architecture, GL-Fusion [32], introduces Structure-Aware Transformers and Graph-Text Cross-Attention to simultaneously process uncompressed text and graph tokens, enabling flexible autoregressive generation alongside graph prediction.

Despite these advancements, two limitations remain. First, many existing methods either linearize graph structures into textual sequences or rely on coarse graph-to-text mappings, which may weaken high-order topology and introduce structural noise into the generation process. Second, most graph-enhanced adapters are not explicitly designed for heterogeneous long-document relations such as syntactic dependency, cross-sentence semantic association, and candidate argument connectivity. These limitations motivate our design of a heterogeneous graph modeling framework with explicit cross-modal alignment and topology-aware structural prompt compression.

3. Materials and Methods

3.1. Overall Architecture

Given a natural language instruction sequence $\mathcal{I}$ for the information extraction task, a source document sequence $\mathcal{D}$, and an offline-constructed heterogeneous document graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, this section aims to leverage explicit document structural knowledge extracted by graph neural networks to enhance and guide the generation process of a large language model (LLM). The proposed method breaks the representational limitation of pure text sequences in one-dimensional space and establishes a structure-enhanced graph-text collaborative generation paradigm, thereby aiming to improve the conditional probability $P_{\Theta}(Y | \mathcal{I}, \mathcal{D}, \mathcal{G})$ of the target structured extraction sequence Y . The overall framework is illustrated in Figure 1.

For clarity, we summarize the main symbols used in the overall framework. Let $\mathcal{I}$ denote the task instruction sequence, $\mathcal{D}$ the source document, and $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ the heterogeneous document graph constructed from $\mathcal{D}$, where $\mathcal{V}$, $\mathcal{E}$, and $\mathcal{R}$ denote the node set, edge set, and relation-type set. The target structured extraction sequence is denoted by Y . Given these inputs, the model aims to maximize the conditional generation probability

$P_{\Theta}(Y | \mathcal{I}, \mathcal{D}, \mathcal{G})$, where Θ denotes the trainable parameters of the whole framework. In the following subsections, $\mathbf{Z}_{graph}$ denotes the graph feature matrix extracted by the MoE Graph Adapter, $\mathbf{Z}'_{graph}$ denotes the aligned graph features after cross-modal projection, and $\mathbf{P}_{struct}$ denotes the fixed-length structural prompt matrix finally injected into the LLM.

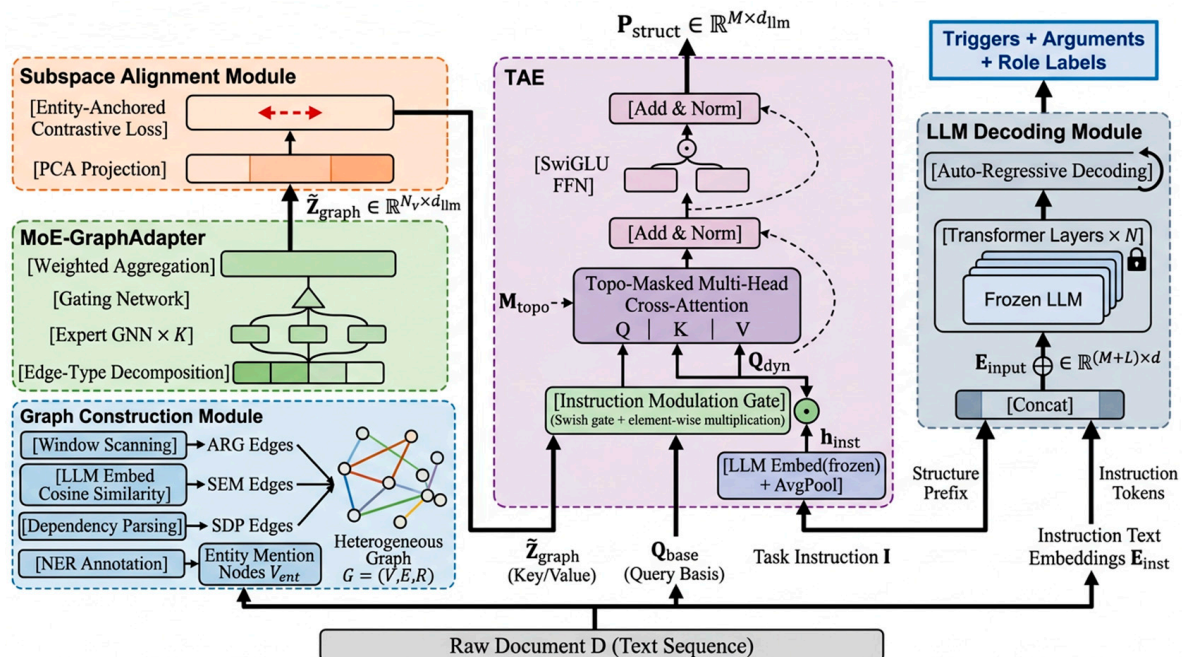


Figure 1. The overall architecture of the proposed method. (**Bottom-Left**) The Graph Construction Module (Section 3.2) builds a heterogeneous graph with Syntactic (SDP), Semantic (SEM), and Argument (ARG) edges using NER and parsing tools. (**Middle-Left**) The MoE-GraphAdapter (Section 3.3) extracts heterogeneous features using dynamic routing. (**Top-Left**) The Subspace Alignment Module (Section 3.4) bridges the modality gap using PCA projection and Dual-Synergy Contrastive Loss (DS³C). (**Right**) The Topology-Aware Encoder (TAE, Section 3.5) compresses graph embeddings into fixed-length structural prompts ($\mathbf{P}_{struct}$) using Topo-Masked Multi-Head Cross-Attention (relying on Queries Q, Keys K, and Values V) and SwiGLU FFN. Finally, these prompts are concatenated with text embeddings for LLM Auto-Regressive Decoding.

The overall computational flow of the model can be abstracted into three core stages:

- **Heterogeneous Feature Extraction.** A mixture-of-experts graph adapter assigned dedicated message-passing parameters to different types of topological relations. It independently extracted and dynamically fused specialized node representations $\mathbf{Z}_{graph}$ from each relation-specific subgraph.
- **Semantic Subspace Alignment.** Through the synergy of dimensional transformation, PCA-based orthogonal projection constraints, and contrastive learning for semantic-spatial alignment, the distributional geometry of the graph features $\mathbf{Z}_{graph}$ was forcibly aligned to the token embedding space of the LLM, yielding the aligned representations $\mathbf{Z}'_{graph}$.
- **Topology-Aware High-Density Feature Compression.** Through task-instruction-modulated dynamic queries and topology-masked cross-attention, the full-graph node features were compressed into fixed-length, high signal-to-noise ratio structural prompt tokens. Finally, the fixed-length structural prompt matrix $\mathbf{P}_{struct}$ was concatenated with the task instructions and the source document in the underlying embedding space, and then fed into the LLM for information extraction.

3.2. Heterogeneous Document Graph Construction

This section describes the construction process of the heterogeneous document graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ from the source document $\mathcal{D}$. The set of heterogeneous edge types is defined as $\mathcal{R} = \{\text{SDP}, \text{SEM}, \text{ARG}\}$, corresponding to syntactic dependencies, cross-sentence semantic associations, and candidate argument relations, respectively.

3.2.1. Node Set Definition

The node set of the graph $\mathcal{V} = \mathcal{V}_{ent} \cup \mathcal{V}_{trig}$ consists of two types of text spans:

- Entity Mention Nodes $\mathcal{V}_{ent}$: A low-rank adaptation fine-tuning Qwen2.5-14B model is used to extract all named entity mentions $\{m_1, \dots, m_{N_e}\}$ from the source document $\mathcal{D}$. Each mention m_i corresponds to a node v_i^{ent} , with attributes including the mention text $\text{text}(m_i)$, entity type $\text{netype}(m_i)$, and the sentence ID $\text{sid}(m_i)$ it belongs to.
- Trigger Candidate Nodes $\mathcal{V}_{trig}$: Based on a predefined trigger vocabulary in the event schema or a low-rank adaptation fine-tuning Qwen2.5-14B model, all trigger candidates $\{t_1, \dots, t_{N_t}\}$ are extracted. Each candidate t_j corresponds to a node v_j^{trig} , with attributes including the trigger text, candidate event type, and sentence ID. The total number of nodes is $N_v = N_e + N_t$.

3.2.2. Heterogeneous Edge Construction

- Syntactic Dependency Path Edges (SDP Edges, E_{sdp})

A dependency parser is applied to each sentence in the document to obtain a semantic dependency tree. For any two nodes $v_i, v_j \in \mathcal{V}$ within the same sentence, an undirected edge is added if the shortest dependency path length between them on the dependency tree does not exceed the threshold L_{sdp} :

$$E_{sdp} = \left\{ (v_i, v_j) \mid \text{sid}(v_i) = \text{sid}(v_j) \wedge d_{dep}(v_i, v_j) \leq L_{sdp} \right\} \quad (1)$$

SDP edges encode intra-sentence word-level semantic modification relations, capturing grammatical structures such as subject–predicate, verb–object, and preposition–object.

- Cross-Sentence Semantic Association Edges (SEM Edges, E_{sem})

Traditional methods rely on external coreference resolution models to build cross-sentence entity links. However, the usability of such models is limited in Chinese scenarios, and prediction errors can propagate to the downstream GNN encoding via graph topology. Therefore, this paper proposes a cross-sentence semantic association edge construction strategy that does not require an external coreference model. Instead, it directly utilizes the LLM embedding layer to calculate the semantic similarity between entity mentions.

For any two entity mention nodes v_i^{ent}, v_j^{ent} located in different sentences, their representations $\mathbf{h}_i^{(0)}, \mathbf{h}_j^{(0)} \in \mathbb{R}^{d_{llm}}$ are obtained using the frozen LLM embedding layer (the specific initialization method is detailed in Section 3.2.1), and the cosine similarity is calculated:

$$s_{i,j}^{sem} = \frac{\mathbf{h}_i^{(0)\top} \mathbf{h}_j^{(0)}}{\|\mathbf{h}_i^{(0)}\| \cdot \|\mathbf{h}_j^{(0)}\|} \quad (2)$$

Here, L_i denotes the number of tokens covered by node v_i , d_{llm} is the embedding dimension of the backbone LLM, and $\mathbf{h}_i^{(0)} \in \mathbb{R}^{d_{llm}}$ is the initial semantic representation of node v_i in the same embedding space as textual tokens.

To control the density of cross-sentence edges, an adaptive threshold based on inter-sentence distance is introduced:

$$\theta_{sem}^{(i,j)} = \theta_{sem}^{base} + \lambda_{decay} \cdot |\text{sid}(v_i) - \text{sid}(v_j)| \quad (3)$$

In Equation (3), θ_{sem}^{base} denotes the base semantic similarity threshold, λ_{decay} is the distance-aware threshold adjustment coefficient, and $\text{sid}(v_i)$ and $\text{sid}(v_j)$ denote the sentence indices of nodes v_i and v_j , respectively. Thus, $|\text{sid}(v_i) - \text{sid}(v_j)|$ measures the inter-sentence distance between the two nodes, and the adaptive threshold becomes more restrictive as the sentence distance increases. A cross-sentence edge is added when $s_{i,j}^{sem} \geq \theta_{sem}^{(i,j)}$. For entity pairs within the same sentence that are not connected by SDP edges, a lower threshold θ_{sem}^{intra} is used to supplement intra-sentence semantic associations. Both parts are combined to form E_{sem} .

This strategy can simultaneously capture two types of cross-sentence entity associations: coreference relations (different mentions of the same entity are naturally highly similar in the LLM embedding space) and semantic-related relations (entity pairs that are not coreferential but semantically close, e.g., location–country, person–organization), the latter of which cannot be covered by traditional coreference resolution.

- Candidate Argument Edges (ARG Edges, E_{arg})

Unlabeled candidate argument edges are established between each trigger candidate node v_j^{trig} and all entity mention nodes within its sentence window W_{arg} :

$$E_{arg} = \left\{ \left(v_j^{trig}, v_i^{ent} \right) \mid |\text{sid}(v_i) - \text{sid}(v_j)| \leq W_{arg} \right\} \quad (4)$$

These edges do not prejudge the argument roles; the specific role assignment is completed by the subsequent LLM generation. The complete edge set is $\mathcal{E} = E_{syn} \cup E_{sem} \cup E_{arg}$.

3.3. Context-Aware MoE Graph Adapter

3.3.1. Semantic Initialization

When receiving graph inputs, traditional GNNs often use random vectors or low-dimensional pre-trained word vectors as node input features. To endow the model with a stronger prior semantic understanding ability, this section extracts the corresponding feature vectors from the LLM's token embedding matrix based on the node's start and end index positions in the text. These vectors are average-pooled to obtain the semantic features of the nodes, serving as the initial features for the corresponding nodes in the graph.

$$\mathbf{h}_i^{(0)} = \frac{1}{L_i} \sum_{l=1}^{L_i} \text{Embed}_{LLM}(w_l^i) \in \mathbb{R}^{d_{llm}} \quad (5)$$

where $\text{Embed}_{LLM}(\cdot)$ represents the vector extracted from the frozen LLM vocabulary matrix. This initialization ensures that the node features reside within the LLM semantic space from the very beginning.

3.3.2. Two-Layer Residual Expert Network

We construct N_{exp} independent expert networks $\{E_1, \dots, E_{N_{exp}}\}$ in parallel, where each expert is a two-layer residual multi-head graph attention network. For expert E_k , the first layer (local neighborhood perception layer) is defined as:

$$\mathbf{z}_{i,k}^{(1)} = \text{ELU} \left(\sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \alpha_{i,j}^{k,(1)} \mathbf{W}_k^{(1)} \mathbf{h}_j^{(0)} \right) \quad (6)$$

The function $\text{ELU}(\cdot)$ denotes the Exponential Linear Unit activation function, which introduces nonlinearity after neighborhood aggregation. $\mathbf{W}_k^{(1)} \in \mathbb{R}^{d_g \times d_{lm}}$ is the learnable linear transformation matrix of the first layer in expert E_k , and $\alpha_{ij}^{k,(1)}$ is the multi-head graph attention weight.

$$\alpha_{ij}^{k,(1)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}_k^{(1)\top} \left[\mathbf{W}_k^{(1)} \mathbf{h}_i^{(0)} \parallel \mathbf{W}_k^{(1)} \mathbf{h}_j^{(0)}\right]\right)\right)}{\sum_{v_l \in \mathcal{N}(v_i) \cup \{v_i\}} \exp\left(\text{LeakyReLU}\left(\mathbf{a}_k^{(1)\top} \left[\mathbf{W}_k^{(1)} \mathbf{h}_i^{(0)} \parallel \mathbf{W}_k^{(1)} \mathbf{h}_l^{(0)}\right]\right)\right)} \quad (7)$$

In Equation (7), $\mathbf{a}_k^{(1)}$ is the learnable attention vector of the first layer in expert E_k , $[\cdot \parallel \cdot]$ denotes vector concatenation, and $(\cdot)^\top$ denotes transpose. The function $\text{LeakyReLU}(\cdot)$ is the leaky rectified linear activation used to compute unnormalized attention scores. The exponential function and the denominator together implement a softmax normalization over all nodes in $\mathcal{N}(v_i) \cup \{v_i\}$, so that the attention weights $\alpha_{ij}^{k,(1)}$ sum to 1 within the local neighborhood.

The second layer introduces residual connections and layer normalization:

$$\mathbf{z}_{i,k}^{(2)} = \text{ELU}\left(\sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \alpha_{ij}^{k,(2)} \mathbf{W}_k^{(2)} \mathbf{z}_{j,k}^{(1)}\right) \quad (8)$$

$$\mathbf{h}_{i,k}^{out} = \text{LayerNorm}\left(\mathbf{z}_{i,k}^{(2)} + \mathbf{z}_{i,k}^{(1)}\right) \quad (9)$$

In Equation (8), $\mathbf{W}_k^{(2)}$ is the learnable transformation matrix of the second layer in expert E_k , and $\alpha_{ij}^{k,(2)}$ is the corresponding second-layer attention weight. $\mathbf{z}_{j,k}^{(1)}$ denotes the intermediate node representation produced by the first layer. Similarly to Equation (6), $\text{ELU}(\cdot)$ is used as the activation function after aggregation. In Equation (9), $\text{LayerNorm}(\cdot)$ denotes layer normalization, which stabilizes training by normalizing the hidden representation. The term $\mathbf{z}_{i,k}^{(2)} + \mathbf{z}_{i,k}^{(1)}$ is a residual connection that preserves first-layer information while facilitating deeper graph representation learning.

3.3.3. Context-Aware Dynamic Routing

We define the global context vector $\mathbf{h}_{ctx}$ as the average pooling of all token embeddings in the input document. The routing probability distribution for node v_i is computed as:

$$\mathbf{g}_i = \text{Softmax}\left(\mathbf{W}_{route} \left[\mathbf{h}_i^{(0)} \parallel \mathbf{h}_{ctx}\right] + \epsilon\right) \quad (10)$$

In Equation (10), $\mathbf{h}_{ctx}$ denotes the global context vector of the input document, obtained by average pooling all token embeddings. $[\mathbf{h}_i^{(0)} \parallel \mathbf{h}_{ctx}]$ denotes the concatenation of the local node representation and the global document context. $\mathbf{W}_{route}$ is the learnable routing matrix, and ϵ is the exploration noise introduced during training to encourage expert diversity. The function $\text{Softmax}(\cdot)$ converts routing logits into a probability distribution over experts, yielding $\mathbf{g}_i$, the routing probability vector for node v_i . The ϵ is the exploration noise during training. Using Top-K sparse routing ($K = 2$), the MoE layer outputs:

$$\mathbf{h}_i^{moe} = \sum_{k \in \text{TopK}(\mathbf{g}_i, K)} g_{i,k} \cdot \mathbf{h}_{i,k}^{out} \quad (11)$$

This produces the full graph feature matrix $\mathbf{Z}_{graph} \in \mathbb{R}^{N_v \times d_g}$.

3.4. Cross-Modal Semantic-Structure Alignment

The GNN extracts rich topological relation features $\mathbf{Z}_{graph}$. However, due to feature aggregation and non-linear mapping during graph convolution, a modality gap and semantic drift exist between this graph feature space and the underlying text representation space of the LLM. If the graph features are directly injected into the LLM, it will introduce cross-modal structural noise, leading to uncontrollable context shifts and extraction hallucinations. Therefore, this section proposes a dual alignment mechanism across the cross-modal semantic space, ranging from macroscopic geometric projection to node feature contrastive learning. Its core is the semantic-structure synergistic alignment loss.

3.4.1. Dimensionality Affine Alignment

As the foundation of graph-text alignment, the graph features $\mathbf{Z}_{graph}$ are first up-projected to the LLM embedding dimension d_{llm} via a linear projection matrix $\mathbf{W}_{up}$:

$$\mathbf{Z}_{proj} = \mathbf{Z}_{graph} \mathbf{W}_{up} + \mathbf{b}_{up} \in \mathbb{R}^{N_v \times d_{llm}} \mathbf{h}_i^{moe} = \sum_{k \in \text{TopK}(\mathbf{g}_i, K)} g_{i,k} \cdot \mathbf{h}_{i,k}^{out} \quad (12)$$

In Equation (12), $\mathbf{Z}_{graph} \in \mathbb{R}^{N_v \times d_g}$ denotes the node feature matrix output by the MoE Graph Adapter, where N_v is the number of graph nodes and d_g is the graph hidden dimension. The learnable matrix $\mathbf{W}_{up}$ projects graph features from d_g to the LLM embedding dimension d_{llm} , and $\mathbf{b}_{up}$ is the corresponding bias term. The resulting matrix $\mathbf{Z}_{proj}$ therefore lies in a space with the same dimensionality as the token embeddings of the backbone LLM.

3.4.2. Semantic Space Orthogonal Projection Constraint

To ensure that the injected graph features reside within the comprehensible domain of the LLM, this section utilizes the intrinsic low-dimensional manifold property of the LLM pre-trained embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d_{llm}}$ for spatial constraints. Principal Component Analysis (PCA) is performed on the LLM token embedding matrix to obtain the principal components $\mathbf{E}$, followed by Singular Value Decomposition (SVD):

$$\mathbf{E} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (13)$$

The right singular vectors corresponding to the top D_{pca} largest singular values are selected to form the principal component basis $\mathbf{C} \in \mathbb{R}^{D_{pca} \times d_{llm}}$. The graph features are then forcibly projected and reconstructed within this subspace:

$$\tilde{\mathbf{Z}}_{graph} = (\mathbf{Z}_{proj} \mathbf{C}^T) \mathbf{C} \in \mathbb{R}^{N_v \times d_{llm}} \mathbf{E} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (14)$$

Here, $\mathbf{E} \in \mathbb{R}^{V \times d_{llm}}$ denotes the frozen token embedding matrix of the backbone LLM, where V is the vocabulary size. After singular value decomposition, the top D_{pca} right singular vectors form the principal component basis $\mathbf{C}$, which defines the dominant semantic subspace of the LLM. Projecting $\mathbf{Z}_{proj}$ into this subspace yields $\tilde{\mathbf{Z}}_{graph}$, which preserves the components most compatible with the LLM semantic geometry while suppressing orthogonal noise components.

This orthogonal projection is designed to filter out components in the graph features that are orthogonal to the LLM semantic space, thereby constraining the graph feature representations within the LLM semantic space and ensuring that the injected graph feature components are comprehensible to the large model.

3.4.3. Semantic-Structure Synergistic Contrastive Learning

The goal of this subsection is to align graph features with textual semantics while preserving the local topology already captured by the graph encoder. To this end, we combine a cross-modal semantic alignment objective with a structure-preserving contrastive objective.

Although PCA orthogonal projection provides global geometric denoising, it does not establish fine-grained semantic bindings between specific graph nodes and their corresponding text tokens. In conventional cross-modal alignment works, the standard InfoNCE loss function is typically employed to achieve this.

However, a single InfoNCE loss exhibits an important topological limitation flaw when processing graph data for IE tasks. When constructing the negative sample repulsion term, standard contrastive learning blindly pushes away all other nodes in the batch (except the anchor point) as negative samples without discrimination. In a heterogeneous document graph, however, strong physical connectivity exists between nodes. If topologically adjacent nodes are blindly pushed apart in the semantic space during the alignment phase, it will lead to severe topological collapse, directly destroying the structural isomorphism just extracted by the GNN.

To address the structural destruction problem caused by the single semantic alignment of standard InfoNCE, this paper proposes a dual contrastive learning mechanism that synergizes cross-modal semantic alignment with graph structure preservation.

- Cross-Modal Semantic Feature Alignment Loss

First, the local semantic pull is established using the correspondence between the entity node $v_i \in \mathcal{V}_{ent}$ and its corresponding text embedding $\mathbf{e}_i^{text}$:

$$\mathcal{L}_{anchor} = -\frac{1}{|\mathcal{V}_{ent}|} \sum_{v_i \in \mathcal{V}_{ent}} \log \frac{\exp(\text{sim}(\bar{\mathbf{z}}_{graph,i}, \mathbf{e}_i^{text}) / \tau_c)}{\sum_{v_j \in \mathcal{V}_{ent}} \exp(\text{sim}(\bar{\mathbf{z}}_{graph,i}, \mathbf{e}_j^{text}) / \tau_c)} \quad (15)$$

where $\mathbf{e}_i^{text} = \frac{1}{L_i} \sum_{l=1}^{L_i} \text{Embed}_{LLM}(w_l^i)$ is the LLM embedding representation of entity v_i , and τ_c is the temperature parameter. PCA projection and contrastive learning synergistically eliminate the graph-text modality gap from both global geometric and local semantic perspectives.

- Graph Structure Consistency Preservation Loss

To prevent the excessive semantic stretching of $\mathcal{L}_{anchor}$ from tearing the topology of the graph, this section introduces a structural consistency contrastive loss. This term mandates that physical neighbors connected in the original graph must strictly maintain their proximity relationship in the latent space after the modality alignment mapping:

$$\mathcal{L}_{struct} = -\frac{1}{|\mathcal{V}|} \sum_{v_i \in \mathcal{V}} \frac{1}{|\mathcal{N}(v_i)|} \sum_{v_j \in \mathcal{N}(v_i)} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j) / \tau_s)}{\sum_{v_k \in \mathcal{V} \setminus \{v_i\}} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k) / \tau_s)} \quad (16)$$

Here, $\mathcal{N}(v_i)$ represents the first-order topological neighborhood set of node v_i in the heterogeneous graph. $\mathbf{z}_j$ is a true neighbor of node v_i (positive sample), and $\mathbf{z}_k$ is an irrelevant node with no edge connection to it in the graph (negative sample). τ_s is the structure contrastive temperature parameter, used to control the compactness of the local connected subgraph.

Compared to the basic InfoNCE loss, it endows the nodes with strong topological cohesiveness, ensuring that the logical skeleton of the interconnected local subgraphs does not distort when migrating as a whole to the text space.

- Joint Optimization Objective

In summary, single cross-modal alignment often comes at the expense of sacrificing the original topological connectivity of the graph. To achieve lossless feature mapping, this paper proposes a joint optimization objective named Dual-Synergy Structure-Semantic Contrastive Loss (DS³C Loss):

$$\mathcal{L}_{DS3C} = \mathcal{L}_{anchor} + \lambda \mathcal{L}_{struct} \quad (17)$$

Parameter description: λ is a dynamic balancing hyperparameter used to find the optimal solution between semantic accuracy and structural fidelity.

Through the above joint optimization paradigm, the model aims to bridge the modality gap at the macro level, outputting aligned features that possess both semantic alignment and structural preservation.

3.5. Topology-Aware Graph-Text Co-Encoder

After MoE-GraphAdapter encoding and subspace alignment, the graph node features are aligned to the LLM's semantic space. However, their dimensionality changes dynamically with the document size. Directly injecting them into the LLM might lead to excessively long contexts and contaminate the original text semantics. Therefore, this section designs a Topology-Aware Encoder (TAE). Its core is the Topology-Guided Sparse Cross-Attention mechanism, which dynamically generates an attention mask based on the topology of the heterogeneous document graph. This restrains the query vector to focus only on graph nodes within its topological neighborhood, compressing the variable-length graph node features into a fixed-length structural prompt matrix, achieving noise filtering and topological locality preservation. The overall structure of TAE is shown in Figure 2.

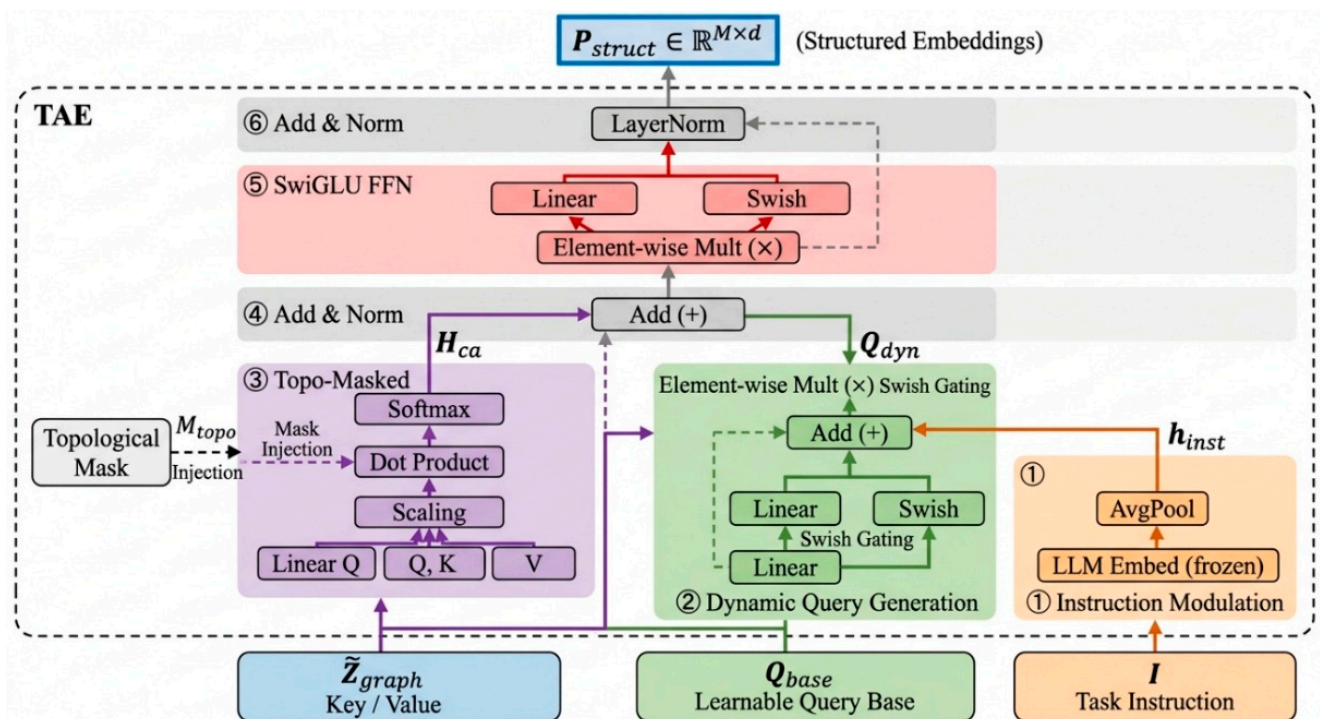


Figure 2. Architecture of the Topology-Aware Encoder (TAE). The left submodule performs topology-masked sparse cross-attention over aligned graph features, the lower-right submodule generates task-instruction-modulated dynamic queries, and the upper submodule applies residual normalization and a SwiGLU feed-forward network to produce the final fixed-length structural prompt matrix P_{struct} . The processing steps are as shown in the diagram.

3.5.1. Task Instruction-Modulated Dynamic Query Generation

A learnable static query basis $\mathbf{Q}_{base} \in \mathbb{R}^{M \times d_{llm}}$ is initialized. Given a task instruction $\mathcal{I}$, an instruction semantic vector $\mathbf{h}_{inst} \in \mathbb{R}^{d_{llm}}$ is obtained via average pooling over the LLM embedding layer. The static basis is affine-modulated:

$$\mathbf{Q}_{dyn} = \mathbf{Q}_{base} + (\mathbf{h}_{inst} \mathbf{W}_{mod}) \odot \text{Swish}(\mathbf{Q}_{base} \mathbf{W}_{gate}) \quad (18)$$

where $\odot$ is element-wise multiplication, and $\text{Swish}(\cdot)$ is the gated activation function. $\mathbf{W}_{mod}$ (instruction modulation projection matrix) and $\mathbf{W}_{gate}$ (content-aware gating projection matrix) are learnable parameters. This modulation makes the queries task-specific, activating different query slots for different extraction intents.

3.5.2. Topology-Masked Sparse Cross-Attention Mechanism

To address the attention dispersion phenomenon caused by global feature injection, this section proposes Topology-Masked Sparse Cross-Attention (TM-SCA). Using $\mathbf{Q}_{dyn}$ as Query and $\bar{\mathbf{Z}}_{graph}$ as Key and Value, a topology mask $\mathbf{M}_{topo} \in \mathbb{R}^{M \times N_v}$ is introduced to perform cross-attention. This ensures each query vector focuses solely on graph nodes within its topological neighborhood, avoiding noise interference from irrelevant nodes.

Construction of the Topology Mask: The topology mask $\mathbf{M}_{topo}$ is dynamically generated based on the heterogeneous document graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constructed in Section 3. The core idea is that nodes topologically closer to event trigger candidates are more likely to carry structural information relevant to the extraction task, while isolated or distant nodes should be masked to reduce noise.

First, identify the set of all event trigger candidate nodes $\mathcal{V}_{trig} \subset \mathcal{V}$ from the heterogeneous document graph. For each node $v_j \in \mathcal{V}$ in the graph, calculate its shortest path distance to the nearest trigger candidate node:

$$d_{min}(v_j) = \min_{v_t \in \mathcal{V}_{trig}} d_{\mathcal{G}}(v_j, v_t) \quad (19)$$

where $d_{\mathcal{G}}(v_j, v_t)$ is the shortest path distance (in terms of number of edges) between node v_j and v_t in graph $\mathcal{G}$. The path calculation spans all edge types (E_{syn} , E_{sem} , E_{arg}). If there is no reachable path between v_j and all trigger candidates, then $d_{min}(v_j) = +\infty$.

Based on the shortest path distance, the elements of the topology mask matrix are defined as:

$$[\mathbf{M}_{topo}]_{m,j} = \begin{cases} 0, & \text{if } d_{min}(v_j) \leq L_{hop} \\ -\infty, & \text{if } d_{min}(v_j) > L_{hop} \text{ or } v_j \text{ is an isolated node} \end{cases} \quad (20)$$

where L_{hop} is a preset topological hop threshold that controls the receptive field of each query vector. When $[\mathbf{M}_{topo}]_{m,j} = -\infty$, the corresponding attention weight approaches zero after Softmax normalization, effectively masking the node from the attention calculation. In the mask matrix, if node v_j is an isolated node or the shortest path distance to all trigger candidates exceeds the constant L_{hop} , then $[\mathbf{M}_{topo}]_{m,j} = -\infty$, otherwise it is 0.

$$\mathbf{H}_{ca} = \text{Softmax} \left(\frac{\mathbf{Q}_{dyn} \bar{\mathbf{Z}}_{graph}^{\top}}{\sqrt{d_k}} + \mathbf{M}_{topo} \right) \bar{\mathbf{Z}}_{graph} \quad (21)$$

The output of the sparse cross-attention goes through a residual connection, layer normalization, and a SwiGLU gated feed-forward network to obtain the output:

$$\mathbf{H}_{norm} = \text{LayerNorm}(\mathbf{H}_{ca} + \mathbf{Q}_{dyn}) \quad (22)$$

$$\mathbf{P}_{struct} = \text{LayerNorm}(\mathbf{H}_{norm} + \text{SwiGLU}(\mathbf{H}_{norm}\mathbf{W}_1) \cdot (\mathbf{H}_{norm}\mathbf{W}_2)) \in \mathbb{R}^{M \times d_{llm}} \quad (23)$$

$\mathbf{P}_{struct}$ serves as the fixed-length, high signal-to-noise ratio structural prompt matrix.

At the input layer of the LLM, the instruction embedding $\mathbf{E}_{\mathcal{I}}$, structural prompt $\mathbf{P}_{struct}$, and document embedding $\mathbf{E}_{\mathcal{D}}$ are concatenated along the sequence dimension:

$$\mathbf{X}_{input} = [\mathbf{E}_{\mathcal{I}} \parallel \mathbf{P}_{struct} \parallel \mathbf{E}_{\mathcal{D}}] \quad (24)$$

Placing the structural prompt between the instruction and the document allows the LLM to naturally attend to topological information via the self-attention mechanism. Because $\mathbf{P}_{struct}$ is spatially aligned, the LLM can interact with it just as it does with standard text tokens.

3.6. Training Strategy

3.6.1. Self-Supervised Modality Alignment Pre-Training

If the MoE-GraphAdapter and alignment modules are directly trained with text generation tasks after parameter initialization, the LLM will dominate the gradients, and the graph encoder will hardly learn useful graph structural knowledge. Therefore, in this stage, the graph encoder is pre-trained independently of the LLM to complete parameter updates.

Parameter Status: Freeze all LLM parameters; optimize only MoE-GraphAdapter, $\mathbf{W}_{up}$, and a Stage-1 exclusive prediction head (Multilayer Perceptron, MLP).

Optimization Objective: Randomly mask partial node features and edges, jointly optimizing four losses:

- Link Prediction Loss ($\mathcal{L}_{link}$)

This is a binary classification loss. For edges that actually exist in the graph (positive samples $\mathbb{E}_{(i,j) \in E_{pos}}$), the predicted probability should approach 1; for randomly sampled non-existent edges (negative samples $\mathbb{E}_{(i,j) \in E_{neg}}$), the predicted probability should approach 0. The expected values of the negative logarithms of the two items are added together. This tests whether the node embeddings encode the true graph topology by predicting the existence of edges.

The MoE output of the node pair is multiplied element-wise and passed through an MLP to predict the edge probability:

$$\hat{p}_{i,j} = \sigma\left(\text{MLP}_{link}(\bar{\mathbf{z}}_{graph,i} \odot \bar{\mathbf{z}}_{graph,j})\right) \quad (25)$$

$$\mathcal{L}_{link} = -\mathbb{E}_{(i,j) \in E_{pos}} [\log \hat{p}_{i,j}] - \mathbb{E}_{(i,j) \in E_{neg}} [\log (1 - \hat{p}_{i,j})] \quad (26)$$

- Semantic-Structure Alignment Loss ($\mathcal{L}_{DS3C}$)

See Equation (17).

- Routing Supervision Loss ($\mathcal{L}_{route}$)

$$\mathcal{L}_{route} = -\frac{1}{N_v} \sum_{v_i \in \mathcal{V}} \log P(\mathbf{g}_i | \mathbf{y}_i^{rel}) \quad (27)$$

Provides explicit classification supervision for the MoE gating network. Pseudo-labels y_i^{rel} are constructed based on the edge type (SDP/SEM/ARG) with the highest participation for each node, requiring the gating distribution $\mathbf{g}_i$ to align with this pseudo-label.

- Load Balancing Loss ($\mathcal{L}_{balance}$)

$$\mathcal{L}_{balance} = N_{exp} \sum_{k=1}^{N_{exp}} f_k \cdot p_k \quad (28)$$

where f_k is the proportion of nodes actually processed by the k -th expert, and p_k is the average probability assigned to the k -th expert by the gating network. This penalizes uneven distribution. If the utilization rate of experts is extremely unbalanced, some experts will not get gradients for a long time, and the parameters cannot be updated, which is equivalent to wasting model capacity.

- Stage 1 Total Loss:

$$\mathcal{L}_{Phase1} = \mathcal{L}_{link} + \mathcal{L}_{DS3C} + \alpha \mathcal{L}_{route} + \beta \mathcal{L}_{balance} \quad (29)$$

where α and β are hyperparameters used to balance the strength of routing decoupling and load balancing, avoiding auxiliary losses from overpowering the main structural learning objectives.

3.6.2. End-to-End Joint Fine-Tuning for Information Extraction

After Stage 1, the graph encoder has a good initialization, but it doesn't yet know what structural information is most important for the IE task. Stage 2 uses real IE annotated data with cross-entropy generation loss as the main objective, allowing gradients to backpropagate from the LLM output end, sequentially passing through the Low-Rank Adaptation (LoRA) adapter, TAE, and MoE, driving the entire chain to align with the goal of "generating correct event structures".

Parameter Status: Freeze LLM backbone; inject LoRA adapter; activate TAE and MoE for end-to-end joint updating.

Joint Optimization Objective:

$$\mathcal{L}_{Phase2} = \mathcal{L}_{CE}(Y | \mathbf{X}_{input}) + \lambda_1 \mathcal{L}_{route} + \lambda_2 \mathcal{L}_{balance} \quad (30)$$

Gradients backpropagate from the LLM output end, sequentially penetrating LoRA, TAE, and MoE, achieving full-chain synergistic optimization driven by a unified semantic objective.

4. Experiment

4.1. Datasets

To comprehensively evaluate the effectiveness and robustness of the proposed method, we conducted experiments on three widely used and challenging benchmark datasets for Information Extraction (IE):

- ACE2005: A standard fine-grained IE benchmark dataset. Following the standard data split (529 documents for training, 74 for validation, and 40 for testing), we focus on the event extraction task, which includes three sub-tasks: event trigger identification and classification, and event argument extraction. To introduce structural information, we construct an intra-document entity co-occurrence graph where nodes represent entities and edges are created if two entities co-occur in the same sentence.
- WikiEvents [13]: A large-scale event extraction dataset where documents and event arguments naturally form a complex graph structure. We strictly adhere to the official

data split provided by the dataset creators. We directly utilize its annotations to construct an event-argument relation graph as the input for the Graph Neural Network (GNN). This dataset serves as an ideal scenario to validate our method's performance on data with explicit, complex graph structures.

- DuEE [14] (Baidu Event Extraction): A large-scale Chinese event extraction dataset derived from real-world web environments, covering broad domains such as life, disasters, and finance. We utilize the official dataset split released by the benchmark. This dataset provides annotations for event type classification, trigger extraction, and argument role classification. It is utilized to test the cross-lingual generalization and robustness of our model.

4.2. Baselines and Comparison Models

We compared our proposed approach with a wide range of state-of-the-art (SOTA) baselines, which can be categorized into the following paradigms:

- Parameter-Efficient Fine-Tuning (PEFT): LLM-LoRA applies Low-Rank Adaptation to the frozen LLM without incorporating any external graph information.
- Instruction-Tuned Unified IE: InstructUIE [4] is a unified IE framework based on Flan-T5 fine-tuned on massive multi-task instruction data;
- Prompt and Code-based IE: GoLLIE [33] enhances zero-shot extraction by aligning LLMs with annotation guidelines using code-style prompts; CodeKGC [17] converts IE into a code generation task to leverage structural constraints; mCODEGPT [34] guides LLMs using hierarchical prompt strategies.
- Agent-based Framework: AgenticIE [35] is an adaptive agent framework that handles complex dependencies in long documents through multi-step reasoning, planning, and self-reflection.
- Graph-Enhanced LLMs: RTE-GMoE [36] utilizes a graph mixture-of-experts mutual learning framework; LayIE-LLM [37] improves extraction via data structuring and output refinement; GaLLa [38] incorporates structural awareness by directly concatenating graph inputs as prompts. GraphAdapter [11] introduces a GNN-based efficient adapter for LLMs.

4.3. Evaluation Metrics

Following standard practices in the IE domain, we employ Precision (P), Recall (R), and the F1-score as our evaluation metrics. The F1-score is defined as the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (31)$$

An extraction is considered correct only when the predicted structured record strictly matches the ground truth. The specific criteria are as follows:

- Relation Extraction (RE): A relation triplet (s, r, o) is correct if and only if the text boundaries and entity types of both the subject s and object o , as well as the relation type r , match the ground truth.
- Event Extraction (EE): A trigger prediction is correct if its text span and predicted event type match the standard annotation.
- Event Argument Extraction (EAE): An argument is correctly extracted if its text span, argument role type, and associated event type all strictly align with the ground truth.

4.4. Implementation Details

Our primary model is built upon the Qwen2.5-14B-Instruct as the base LLM. The Mixture-of-Experts (MoE) Graph Adapter utilizes $N_{exp} = 4$ parallel two-layer Graph

Attention Networks (GAT) with a hidden dimension $d_g = 768$. The topological mask hop threshold is set to $L_{hop} = 3$, and the maximum shortest dependency path (SDP) length is $L_{sdp} = 4$.

To ensure a fair and reproducible comparative evaluation, all baseline methods were trained and evaluated using the same standard data split protocols across the three datasets. For LLM-based baselines (e.g., LLM-LoRA, GoLLIE, GraphAdapter), we standardized the training configurations where applicable: models were trained for up to 10 epochs using the AdamW optimizer with a batch size of 8 and a linear learning rate decay schedule starting from 5×10^{-5} . Early stopping was applied with a patience of 3 epochs based on validation F1-score to prevent overfitting. For prompt-based and agent-based baselines (e.g., InstructUIE, AgentIE), we strictly followed the original task-specific prompt templates and data preprocessing pipelines provided in their official implementations to maximize their intended performance.

During the fine-tuning phase, we employ the AdamW optimizer with a learning rate of 5×10^{-5} . The initial learning rate is set to 1×10^{-4} , with a cosine decay schedule and a warmup ratio of 0.06. The per-device batch size is 2, and gradient accumulation steps are set to 8, resulting in an effective batch size of 16. All experiments were conducted on a workstation equipped with 2×NVIDIA RTX A6000 (48 GB) GPUs (NVIDIA, Santa Clara, CA, USA). Detailed hyperparameters are summarized in Table 1.

Table 1. Hyperparameters configurations.

Parameters	Value	Description
L_{sdp}	4	maximum shortest dependency-path distance allowed when constructing syntactic dependency edges.
θ_{sem}	0.85	base semantic similarity threshold used for building semantic association edges between entity nodes
W_{arg}	3	sentence-window size used to connect trigger candidates with nearby entity mentions.
N_{exp}	4	number of experts in the MoE Graph Adapter.
d_g	768	hidden dimension of the graph encoder output.
K	2	number of selected experts in Top-K sparse routing.
σ^2	0.01	variance of the exploration noise used in the routing process during training.
D_{pca}	1024	number of principal components retained in the PCA-based semantic subspace projection.
τ_c	0.07	temperature parameter for the cross-modal semantic contrastive loss.
τ_s	0.1	temperature parameter for the structure-preserving contrastive loss.
λ	0.5	balancing coefficient between semantic alignment and structural consistency in the DS ³ C loss.
M	128	number of structural prompt slots generated by the TAE.
L_{hop}	3	topological hop threshold used in the topology mask of the TAE.
α, λ_1	0.1	weights of the routing supervision loss in Phase 1 and Phase 2 training, respectively.
β, λ_2	0.1	weights of the load balancing loss in Phase 1 and Phase 2 training, respectively.

5. Result and Discussion

5.1. Main Results

Tables 2–4 present the comparative performance of our proposed method against various baselines on the ACE2005, WikiEvents, and DuEE downstream tasks, respectively. For LLM-based baselines, to ensure a fair comparison, we reimplemented or reran these methods under the same experimental environment using Qwen2.5-14B as the backbone model whenever applicable, with the same hardware setting, training protocol, and evaluation pipeline as our method.

Table 2. Performance on ACE2005 datasets.

Method	RE	EE	EAE
LLM-LoRA	68.5	65.5	56.5
CodeKGC	70.0	73.1	60.2
InstructUIE	73.2	72.8	59.8
GoLLIE	75.5	74.2	61.5
RTE-GMoE	71.2	72.3	58.2
LayIE-LLM	76.1	73.2	62.1
mCODEGPT	72.1	68.6	57.9
AgentIE	75.2	73.5	63.0
GaLLa	71.2	69.8	62.1
GraphAdapter	75.6	74.1	64.3
Ours	79.1 ± 0.2	74.8 ± 0.3	66.2 ± 0.1

Table 3. Performance on WikiEvents datasets.

Method	EE	EAE
LLM-LoRA	64.2	45.1
CodeKGC	68.1	51.8
InstructUIE	68.5	55.5
GoLLIE	69.1	55.8
RTE-GMoE	64.1	52.3
LayIE-LLM	70.2	58.1
mCODEGPT	68.4	52.1
AgentIE	73.3	57.5
GaLLa	70.4	55.6
GraphAdapter	71.2	58.5
Ours	72.5 ± 0.2	60.4 ± 0.5

Table 4. Performance on DuEE datasets.

Method	EE	EAE
LLM-LoRA	60.5	48.2
CodeKGC	65.5	52.0
InstructUIE	71.2	51.5
GoLLIE	70.5	54.0
RTE-GMoE	68.4	52.2
LayIE-LLM	72.8	55.8
mCODEGPT	67.4	50.2
AgentIE	70.5	56.0
GaLLa	68.1	52.9
GraphAdapter	73.1	54.7
Ours	75.1 ± 0.1	58.2 ± 0.3

5.2. Discussion and Analysis

Based on the experimental results, our method achieves consistently competitive performance across the evaluated datasets and tasks, with the largest gains appearing in structure-sensitive settings. We interpret these improvements as evidence that explicit structural signals can complement sequence-based LLM representations in information extraction.

Compared to meticulously instruction-tuned models (e.g., InstructUIE, GoLLIE) and agent-based methods with complex reasoning capabilities (e.g., AgentIE), our approach consistently achieves higher F1-scores. These performance gains are particularly noticeable in Relation Extraction (RE) and Event Argument Extraction (EAE) tasks, which heavily rely

on text topology and long-distance dependencies. For instance, on the ACE2005 dataset, our model achieves F1-scores of 79.1% and 66.2% in RE and EAE, outperforming the strongest respective baselines (LayIE-LLM and AgentIE) by an absolute margin of 3.0% and 3.2%. This suggests that explicitly injecting and aligning heterogeneous graph structures can help alleviate the “structural blindness” often observed in LLMs when inferring complex argument relationships.

The WikiEvents dataset is characterized by long documents and intricate argument roles. In the globally structure-dependent EAE task, our method (60.4%) outperforms both basic instruction-tuned models and multi-step reasoning frameworks like AgentIE (57.5%). This indicates the potential utility of our MoE Graph Adapter and Topology-Aware Encoder (TAE) in crossing sentence boundaries and capturing entity coreferences and long-range syntactic dependencies within lengthy contexts.

In the DuEE dataset, a real-world Chinese event extraction corpus, our framework still yields the highest F1-scores among the evaluated models (75.1% in EE and 58.2% in EAE). Given that Chinese syntactic structures (such as omission and inversion) are often more implicit than English, these results suggest that our mechanism for extracting and fusing syntactic trees and semantic correlation graphs exhibits a degree of cross-lingual generalizability.

Compared to basic fine-tuned models (LLM-LoRA): Our framework yields substantial improvements across all metrics (e.g., a notable +9.7% increase in ACE2005 EAE). This highlights the limitations of purely 1D sequence autoregressive models in handling non-Euclidean topologies, supporting the motivation for introducing external structural knowledge.

Compared to Code-based/Prompt-based models (CodeKGC, mCODEGPT): Although these methods attempt to apply structural constraints via the LLM’s code generation capabilities, they essentially “linearize” graph structures into text. The observed improvements suggest that the proposed PCA orthogonal projection and dual contrastive learning—which aim to align graph features to the underlying semantic space more faithfully—can be more effective than simple sequence concatenation in preserving high-order topological information.

Compared to other Graph-enhanced LLMs (RTE-GMoE, GaLLa): Our method also achieves better performance than other models that introduce GNNs. We hypothesize this is partly due to our proposed “Dual-Synergy Structure-Semantic Contrastive Strategy,” which aims to reduce the modality gap and structural noise often introduced by directly injecting graph features, helping the LLM utilize graph knowledge while maintaining its original language generation capabilities.

5.3. Generalization Across Different LLM Backbones

To verify the model-agnostic nature and scalability of our method, we integrated our framework into various LLM backbones with different scales and architectures. These include dense models Qwen2.5-7B, Llama2-7B/13B, Llama3-8B and an MoE-based model DeepSeek-V2-Lite, activating only 2.4B parameters per token during inference. The comparison against their respective baseline fine-tuned versions is shown in Table 5.

The results in Table 5 indicate that our method consistently yields substantial performance gains (ranging from +7.2% to +9.3% on average) over pure LoRA fine-tuning, regardless of the base model’s size or internal architecture. This confirms that our approach can be seamlessly integrated into diverse LLMs, stably exerting its advantage in structural knowledge injection.

Table 5. Generalization analysis on different LLM backbones.

Basemodel	Method	ACE05-Avg	WikiEvents-Avg	Absolute Gain
Llama2-7B	LoRA	49.5	36.5	-
Llama2-7B	Ours	56.2	45.8	+8.0
Llama2-13B	LoRA	59.8	42.2	-
Llama2-13B	Ours	63.5	53.5	+7.5
Qwen2.5-7B	LLM-LoRA	58.2	49.5	-
Qwen2.5-7B	Ours	61.8	60.2	+7.2
Llama3-8B	LoRA	55.8	48.1	-
Llama3-8B	Ours	62.4	60.1	+9.3
DeepSeek-V2-Lite	LLM-	68.4	66.1	-
DeepSeek-V2-Lite	Ours	75.2	73.7	+7.2

5.4. Ablation Study

To deeply investigate the contribution of each architectural component, we conducted a rigorous ablation study on both datasets. The results are summarized in Table 6.

Table 6. Ablation study on the method component.

Category	Variant	Description	ACE2005	WikiEvents	AvgDrop
Full Model	-	Our complete framework	72.7	66.45	-
Pre-training & Architecture	w/oPT	No pre-training Stage	70.50	63.10	−2.78
	w/o ctx	Using only linear routing	68.53	60.72	−4.95
	w/oMOE:	Only single 2-layer GAT	70.59	63.01	−2.78
	w/oTAE	No TAE	69.80	58.90	−5.23
Alignment Mechanism	w/o Contrastive	No contrastive learning	71.30	64.25	−1.80
	w/o PCA	Removes the PCA constraint	71.70	64.85	−1.30
	Linear Only	No PCA/Contrastive	70.79	63.51	−2.43
Graph Signals	w/o SDP	Graph without SDP edges	71.29	65.06	−1.40
	w/o SEM	Graph without SEM edges	70.87	64.68	−1.80
	w/o ARG	Graph without ARG edges	70.48	64.23	−2.20
Prompt	Text-described Graph	Natural language description of graph in text prompt.	68.40	59.65	−5.55
	Soft Prompt	Using a learnable soft prompt instead of graph module	68.24	57.986	−6.53
	Text Prompt	Only natural language instruction	68.13	56.77	−7.12
	LORA	Baseline LoRA.	66.52	54.63	−9.00

Category 1 examines the contribution of the pre-training and architectural components. Removing self-supervised pre-training (w/o PT) reduces performance by 2.78 points on average, suggesting that pre-training provides a more favorable initialization for subsequent task-specific optimization. Replacing context-aware routing with a simpler linear routing scheme (w/o ctx) leads to a larger degradation of 4.95 points, indicating that input-dependent routing is beneficial under our experimental setting. Replacing the MoE Graph Adapter with a single shared two-layer GAT (w/o MoE) also degrades performance by 2.78 points. Taken together, these results show that the proposed routing and multi-expert design are useful for heterogeneous-graph encoding. However, the present ablation results do not by themselves establish a strict functional disentanglement of different topological signals across experts; demonstrating such specialization would

require additional analyses such as expert-assignment statistics, routing entropy analysis, or routing-behavior visualization.

Category 2 evaluates the cross-modal alignment mechanism. All variants retain the same linear up-projection to match the LLM hidden dimensionality. Removing both PCA-based projection and contrastive learning (Linear Only) results in a 2.43-point average drop. Removing only PCA (w/o PCA) yields a 1.30-point decrease, while removing only contrastive learning (w/o Contrastive) causes a 1.80-point decrease. These results indicate that both components contribute to the final performance, with contrastive learning showing a somewhat larger impact in the current setting. More cautiously, the results support the view that the two components are complementary within the present framework. They do not yet provide direct mechanistic evidence for a strong synergistic interaction beyond performance complementarity.

Category 3 analyzes the contribution of different graph signals. Removing SDP, SEM, and ARG edges causes average drops of 1.40, 1.80, and 2.20 points, respectively, which indicates that all three relation types contribute to the final extraction performance. Within the current experimental setting, the larger decrease observed for removing ARG edges suggests that trigger-argument connectivity provides particularly useful supervision for downstream decoding. The effect of SEM edges also supports the value of cross-sentence semantic associations, while SDP edges remain helpful for capturing intra-sentence structural dependencies.

At the same time, these results should be interpreted with some caution. The current single-factor ablations show that each structural signal matters, but they do not fully disentangle the interaction among SDP, SEM, and ARG edges, nor do they establish that one signal is universally the most important under all document lengths, event types, or data regimes. A more complete understanding would require conditional analyses by document complexity, event category, or edge-density setting. We therefore interpret these results as evidence that the three graph signals are complementary components of the heterogeneous document graph rather than as a definitive ranking of structural importance.

Finally, Category 4 investigates whether the gains mainly come from architectural design or prompt engineering. We observe a clear progression: a Text-only Prompt improves over the Vanilla LLM, but remains inferior to methods that incorporate structural information. Injecting graph structure as a Text-described Graph also improves performance, showing that the structure itself is useful even when linearized into text. However, both this variant and Soft Prompt Only remain worse than the full model. This suggests that directly injecting graph features with proper alignment is more effective than representing structural knowledge only as a 1D prompt sequence.

5.5. Error Analysis

To meticulously diagnose the limitations of our model and understand the underlying reasons for its performance gains, we conducted a comprehensive error analysis. We randomly sampled incorrect extraction cases from the validation set and compared the error distributions between our proposed method and the strongest generative baseline, GoLLIE.

We categorized the extraction errors into four distinct types:

- **Missing Argument:** The model completely fails to identify and extract an existing argument.
- **Wrong Boundary:** The model identifies the correct core entity, but the extracted text span (boundaries) does not strictly match the ground truth (e.g., extracting “the police officer” instead of “police”).
- **Role Confusion:** The model correctly extracts the argument span but assigns it to the wrong event role.

- **Cross-Sentence Failure:** The model fails to capture arguments that reside in a different sentence from the event trigger.

Figure 3 illustrates the relative distribution of these major error types within the remaining error pool. Based on the analysis of these specific cases, several critical observations can be drawn:

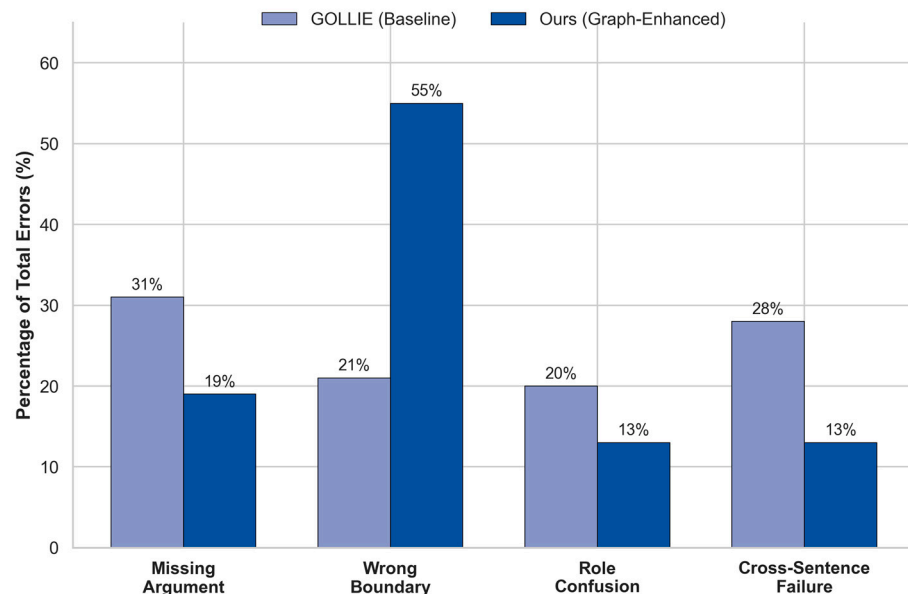


Figure 3. Comparison of Error Situations.

Drastic Mitigation of Topology-dependent Errors: Crucially, it must be noted that the absolute total number of errors produced by our method was substantially reduced compared to the GoLLIE baseline. Specifically, the relative proportions of Cross-Sentence Failure (from 28% to 13%) and Role Confusion (from 20% to 13%) experienced the most significant declines. GoLLIE, relying solely on 1D sequence reasoning, frequently struggles with long-distance dependencies. In contrast, our method explicitly injects a heterogeneous document graph (incorporating inter-sentence semantic edges and shortest dependency paths) into the LLM. This explicitly bridges the spatial gap between triggers and distant arguments, effectively resolving the “heavy hitters” among extraction errors.

The Shift in Error Proportions: At first glance, the proportion of Wrong Boundary errors in our method appears to surge to 55% (up from 21% in the baseline). However, a deeper inspection of the validation cases reveals that this is a relative statistical shift rather than an absolute degradation.

Because our framework successfully resolved the majority of complete extraction failures (Cross-Sentence and Role Confusion), those intractable cases were largely eliminated from the error pool. Consequently, cases involving subtle boundary mismatches became the dominant remaining issue. Furthermore, many arguments that GoLLIE completely failed to find (previously classified as Missing Argument) are now successfully located by our graph-enhanced LLM, but occasionally with minor token-level boundary deviations (e.g., including extraneous adjectives). This essentially downgrades a severe Missing error into a lighter Wrong Boundary error.

Remaining Challenges in Implicit Extraction: While the Missing Argument rate dropped from 31% to 19%, it remains a challenge. Error analysis shows that the remaining missing arguments are predominantly highly implicit entities or pronouns that lack clear syntactic dependencies or lexical similarity within the document. Addressing these deep semantic omissions, possibly through more fine-grained coreference resolution mechanisms or advanced LLM decoding strategies, will be a focus of our future work.

In summary, the error distribution shift supports our initial hypothesis: explicitly aligning structural knowledge with LLMs substantially alleviates high-order topological failures, leaving only localized, fine-grained span alignments as the primary remaining frontier.

5.6. Efficiency and Robust Analysis

5.6.1. Computational Cost and Efficiency

To assess the computational overhead introduced by the proposed structural components, we report the efficiency statistics in Table 7.

Table 7. Computational Cost and Efficiency Comparison.

Method	Trainable Params	Total Params	GPU Memory (Inference)	Training Time (Hour)	Inference Latency (ms/Token)
LLM-LoRA	45.8 M	14.7 B	29.5 GB	9.5	24.2
Ours	182.5 M	14.8 B	30.2 GB	12.2	25.8

Note: Evaluated on a workstation with NVIDIA RTX A6000 (48 GB) GPUs. GPUs using bf16 precision. Basemodel Qwen2.5-14B.

From a cost–benefit perspective, the full framework introduces additional trainable modules beyond vanilla LoRA, mainly through the MoE Graph Adapter and the TAE, while also requiring document-graph construction and cross-modal alignment. This increases training complexity and modestly increases inference memory and latency relative to simpler baselines. Nevertheless, the empirical gains are concentrated in structure-sensitive settings such as relation extraction and event argument extraction, where long-range dependencies and cross-sentence interactions are more prominent. We therefore view the full design as most appropriate for scenarios in which extraction quality on structurally complex documents is the primary objective and a moderate computational overhead is acceptable.

5.6.2. Scalability on Long Documents

Long-document IE places substantial demands on context scalability. As shown in Table 8, when the input length increases from 1000 to 8000 tokens, the average number of graph nodes grows approximately linearly, and the graph construction time increases accordingly. However, the GPU memory usage and decoding latency of the overall system increase much more moderately. This favorable scalability is largely enabled by the Topology-Aware Encoder, which compresses variable-size graph representations into a fixed-length structural prompt.

Table 8. Scalability Across Different Document Lengths.

Doc Length (Tokens)	Graph Nodes (Avg)	Graph Const Time	GPU Memory (Inference)
1000	146.4	0.12 s/doc	30.2 GB
2000	276.1	0.25 s/doc	31.5 GB
4000	559.4	0.58 s/doc	34.1 GB
8000	976.1	1.45 s/doc	38.6 GB

Note: Results are measured under the same hardware setting as Table 7.

As a result, although the underlying document graph becomes larger for longer inputs, the additional context burden introduced to the LLM remains bounded, allowing the model to handle long documents without prohibitive memory growth or latency overhead.

5.6.3. Robustness Analysis Under Structural Noise

Our heterogeneous graph is partially constructed using outputs from the spaCy (version 3.7.2) parser. To evaluate the model's sensitivity to preprocessing noise, we inject controlled structural perturbations into the graph, as shown in Table 9. Randomly deleting 10% of edges, which simulates recall failures in the preprocessing pipeline, leads to only a 0.55% F1 drop. Adding 10% random noisy edges, which simulates incorrect parser predictions, results in a slightly larger but still limited drop of 0.85%. Overall, the model remains relatively stable under moderate structural noise. A possible explanation is that the sparsity-aware graph interaction mechanism can mitigate the impact of irrelevant perturbations, while the underlying LLM still provides complementary semantic cues when part of the structure is missing or noisy. These results suggest that the proposed method is reasonably robust to realistic preprocessing imperfections.

Table 9. Result of Robustness Analysis.

Noise Ratio	Random Edge Deletion	Random Edge Addition
0% (clean)	72.70	72.70
5%	72.55 (−0.15)	72.48 (−0.22)
10%	72.15 (−0.55)	71.85 (−0.85)
20%	71.40 (−1.30)	70.65 (−2.05)

Note: Structural noise is injected by randomly deleting or adding edges in the heterogeneous document graph. Results are reported on ACE2005 F1. Values in parentheses denote the absolute drop relative to the clean setting.

5.7. Case Study

To further examine the practical effect of the proposed framework, we present two representative case studies from relation extraction (RE) and event extraction (EE). Unlike purely sequence-based methods, our approach integrates heterogeneous structural signals through the document graph and injects them into the LLM via aligned structural prompts, which is particularly beneficial in structurally complex extraction scenarios. The detailed results are presented in Table 10.

Table 10. Case Study Examples.

Case Type	Sample
RE: entities belonging to two relations	As the JAMA article was being released, three dozen children in Delhi, India, began to die from acute renal failure at two hospitals.
EE: predicate in a clause	Clinton was deeply affected by the loss of the 19 Rangers, 18 of who died on October 3rd, with Matt Reersen passing away three days later.

The first case comes from relation extraction, where the same entity pair participates in two different relations. In the sentence, “India” and “Delhi” simultaneously express a geographical inclusion relation and an administrative affiliation relation. Such cases are challenging because a single entity pair may simultaneously participate in multiple fine-grained relations, which increases the difficulty of relation discrimination and relation direction modeling. In our framework, these multi-type associations are explicitly represented in the heterogeneous document graph and then aligned to the LLM semantic space before injection. This enables the model to better distinguish fine-grained relation semantics and more robustly handle relation directionality.

The second case comes from event extraction. The trigger word “died” is associated with multiple event elements, including the victim “19 Rangers”, the related person “Clinton”, and the temporal expressions “October 3rd” and “three days later”. The difficulty of this example lies in the fact that the event arguments are distributed across different parts of the sentence and require the model to capture both long-distance dependencies and

structurally relevant context. Our method addresses this challenge by establishing explicit trigger–argument connections through candidate argument edges, while syntactic dependency edges and semantic association edges further supplement the structural context. Moreover, the topology-masked sparse cross-attention in the TAE enables the model to focus on graph nodes that are topologically more relevant to the trigger, thereby reducing interference from irrelevant context and improving event extraction completeness.

Overall, these two cases demonstrate that the proposed framework is effective in both multi-relation discrimination and complex event argument extraction. They further confirm that the combination of heterogeneous graph modeling, cross-modal alignment, and topology-aware structural prompt compression helps alleviate the structural blind spots of LLMs in long-document information extraction.

6. Conclusions

This paper presents a generative information extraction framework that integrates heterogeneous graph modeling, cross-modal alignment, and topology-aware structural prompting for long-document information extraction. Across ACE2005, WikiEvents, and DuEE, the method yields consistent performance improvements over the compared base-lines, particularly on tasks that rely on cross-sentence or structurally complex evidence. The ablation results further indicate that the graph encoder, alignment module, and structural prompting module each contribute to the final performance. At the same time, the current experiments mainly support the usefulness of these components at the empirical level. More fine-grained evidence would be needed to make stronger mechanistic claims about expert specialization, topology preservation, or cross-module interaction. Future work will therefore focus on improving experimental transparency, adding targeted interpretability analyses, and exploring lighter variants of the framework for more resource-constrained deployment.

Author Contributions: Conceptualization, Y.L. and C.T.; methodology, Y.L.; software, Y.L.; validation, C.T. and B.Z.; formal analysis, C.T.; investigation, B.Z.; resources, W.Z.; data curation, B.Z.; writing—original draft preparation, Y.L.; writing—review and editing, C.T. and B.Z.; visualization, Y.L.; supervision, W.Z.; project administration, W.Z.; funding acquisition, W.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China under Grant (NO. 62072469), Shandong Data Open Innovative Application Laboratory, and the Natural Science Foundation of Shandong Province under Grant (NO. ZR2023MF041).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and models generated during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gao, Q.; You, H. Research survey on event extraction technology. *Inf. Stud. Theory Appl.* **2013**, *36*, 5.
2. Zhu, L.; Zheng, H. Biomedical event extraction with a novel combination strategy based on hybrid deep neural networks. *BMC Bioinform.* **2020**, *21*, 47. [[CrossRef](#)] [[PubMed](#)]
3. Xu, D.; Chen, W.; Peng, W.; Zhang, C.; Xu, T.; Zhao, X.; Wu, X.; Zheng, Y.; Chen, E. Large language models for generative information extraction: A survey. *arXiv* **2023**, arXiv:2312.17617. [[CrossRef](#)]
4. Wang, X.; Zhou, W.; Zu, C.; Xia, H.; Chen, T.; Zhang, Y.; Zheng, R.; Nie, J.; Li, P.; Zhang, Q.; et al. InstructUIE: Multi-task instruction tuning for unified information extraction. *arXiv* **2023**, arXiv:2304.08085.

5. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24824–24837.
6. Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the middle: How language models use long contexts. *Trans. Assoc. Comput. Linguist.* **2024**, *12*, 157–173. [\[CrossRef\]](#)
7. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [\[CrossRef\]](#)
8. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [\[CrossRef\]](#)
9. Liang, W.; Zhang, Y.; Kwon, Y.; Yeung, S.; Zou, J. Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 17612–17625.
10. Luo, Y.; Yang, Z.; Meng, F.; Li, Y.; Zhou, J.; Zhang, Y. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv* **2023**, arXiv:2308.08747. [\[CrossRef\]](#)
11. Huang, X.; Han, K.; Yang, Y.; Bao, D.; Tao, Q.; Chai, Z.; Zhu, Q. Can GNN be good adapter for LLMs? In Proceedings of the ACM Web Conference 2024 (WWW '24), Singapore, 13–17 May 2024; pp. 1–12.
12. Wang, D.; Zuo, Y.; Li, F.; Wu, J. LLMs as zero-shot graph learners: Alignment of GNN representations with LLM token embeddings. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 5950–5973.
13. Li, S.; Ji, H.; Han, J. Document-level event argument extraction by conditional generation. *arXiv* **2021**, arXiv:2104.05919. [\[CrossRef\]](#)
14. Li, X.; Li, F.; Pan, L.; Chen, Y.; Peng, W.; Wang, Q.; Lyu, Y.; Zhu, Y. DuEE: A large-scale dataset for Chinese event extraction in real-world scenarios. In *Natural Language Processing and Chinese Computing, Proceedings of the 9th CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC 2020), Zhengzhou, China, 14–18 October 2020; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12431, pp. 534–545.*
15. Li, J.; Sun, A.; Han, J.; Li, C. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [\[CrossRef\]](#)
16. Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; Wu, H. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 5755–5772.*
17. Bi, Z.; Chen, J.; Jiang, Y.; Xiong, F.; Guo, W.; Chen, H.; Zhang, N. CodeKGC: Code language model for generative knowledge graph construction. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2024**, *23*, 45. [\[CrossRef\]](#)
18. Xu, J.; Sun, M.; Zhang, Z.; Zhou, J. ChatUIE: Exploring chat-based unified information extraction using large language models. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Torino, Italy, 20–25 May 2024; pp. 3146–3152.
19. Li, Z.; Wang, S.; Fang, R.; Bao, M.; Wu, Z.; Song, S.; Li, Y.; He, Z. MR-UIE: Multi-perspective reasoning with reinforcement learning for universal information extraction. *arXiv* **2025**, arXiv:2509.09082. [\[CrossRef\]](#)
20. Ma, Y.; Wang, Z.; Cao, Y.; Li, M.; Chen, M.; Wang, K.; Shao, J. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Association for Computational Linguistics: Stroudsburg, PA, USA, 2022; pp. 6759–6774.*
21. Du, X.; Cardie, C. Event extraction by answering (almost) natural questions. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 671–683.
22. Liu, J.; Yang, C.; Lu, Z.; Chen, J.; Li, Y.; Zhang, M.; Bai, T.; Fang, Y.; Sun, L.; Zheng, V.W.; et al. Towards graph foundation models: A survey and beyond. *arXiv* **2023**, arXiv:2310.11829. [\[CrossRef\]](#)
23. Yan, H.; Jin, X.; Meng, X.; Guo, J.; Cheng, X. Event detection with multi-order graph convolution and aggregated attention. 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), Hong Kong, China, 3–7 November 2019; pp. 5766–5770.
24. Cui, S.; Yu, B.; Liu, T.; Zhang, Z.; Wang, X.; Shi, J. Edge-enhanced graph convolution networks for event detection with syntactic relation. *arXiv* **2020**, arXiv:2002.10757. [\[CrossRef\]](#)
25. Wadden, D.; Wennberg, U.; Luan, Y.; Hajishirzi, H. Entity, relation, and event extraction with contextualized span representations. 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), Hong Kong, China, 3–7 November 2019; pp. 5784–5789.
26. Nasar, Z.; Jaffry, S.W.; Malik, M.K. Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.* **2021**, *54*, 20. [\[CrossRef\]](#)
27. Zhu, X.; Yang, C.; Wang, Q.; Guo, Z.; Wang, Y.; Liu, Z.; Wang, C.; Shi, C. Graph4LLM: A systematic survey of graph-enhanced large language models. *TechRxiv*, 13 February 2026. [\[CrossRef\]](#)
28. Liu, Z.; Yu, X.; Fang, Y.; Zhang, X. GraphPrompt: Unifying pre-training and downstream tasks for graph neural networks. In Proceedings of the ACM Web Conference 2023, Austin, TX, USA, 30 April–4 May 2023; pp. 417–428.

29. Chen, R.; Zhao, T.; Jaiswal, A.; Shah, N.; Wang, Z. LLaGA: Large language and graph assistant. In Proceedings of the International Conference on Machine Learning (ICML), Vienna, Austria, 21–27 July 2024.
30. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Larson, J. From local to global: A graph RAG approach to query-focused summarization. *arXiv* **2024**, arXiv:2404.16130. [[CrossRef](#)]
31. Li, S.; He, Y.; Guo, H.; Zheng, Y.; Qiu, X.; Zhang, C.; Zhang, Y. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, FL, USA, 12–16 November 2024*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024.
32. Yang, H.; Wang, X.; Tao, Q.; Hu, S.; Lin, Z.; Zhang, M. GL-Fusion: Rethinking the combination of graph neural network and large language model. *arXiv* **2024**, arXiv:2412.06849. [[CrossRef](#)]
33. Sainz, O.; García-Ferrero, I.; Agerri, R.; Lopez de Lacalle, O.; Rigau, G.; Agirre, E. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. In Proceedings of the 12th International Conference on Learning Representations (ICLR 2024), Vienna, Austria, 7–11 May 2024.
34. Zhang, K.; Huang, T.; Malin, B.A.; Osterman, T.; Long, Q.; Jiang, X. Introducing mCODEGPT as a zero-shot information extraction from clinical free text data tool for cancer research. *Commun. Med.* **2025**, *5*, 422. [[CrossRef](#)] [[PubMed](#)]
35. Colakoglu, G.; Gürkan, S.; Jonathan, F. AgenticIE: An Adaptive Agent for Information Extraction from Complex Regulatory Documents. *arXiv* **2025**, arXiv:2509.11773.
36. Wulamu, A.; Gong, K.; Zhengyu, L.; Han, Y.; Zhu, Z.; Xing, B. RTE-GMoE: A Model-agnostic Approach for Relation Triplet Extraction via Graph-based Mixture-of-Expert Mutual Learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Suzhou, China, 4–9 November 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025.
37. Colakoglu, G.; Gürkan, S.; Jonathan, F. Problem solved? information extraction design space for layout-rich documents using llms. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, 4–9 November 2025*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025; pp. 17908–17927.
38. Zhang, Z.; Yu, H.; Lee, S.; Di, P.; Li, J.; Wang, R. Galla: Graph aligned large language models for improved source code understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vienna, Austria, 27 July–1 August 2025; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.