IntuitiveGraphLLM: Intuitive Graph-based Text Representation with Large Language Model

Md Mostafizer Rahman[1, 2], **Yutaka Watanobe**[3], **Syed Rameez Naqvi**[1], **Lu Peng**[1]

[1]Tulane University, New Orleans, USA
[2]Lucy Family Institute for Data & Society, University of Notre Dame, IN, USA
[3]University of Aizu, Aizuwakamatsu, Japan
mrahman9@tulane.edu, yutaka@u-aizu.ac.jp, snaqvi@tulane.edu, lpeng3@tulane.edu

Abstract

Graphical representations of text can sharpen the inductive biases of large language models (LLMs), yet most graph-based approaches rely on co-occurrence, order, or position alone and therefore over-connect unrelated tokens while missing conceptually salient links. We introduce Intuitive Graphs (IGs)—graphs that explicitly encode both (i) structural context (local order/proximity/position) and (ii) conceptual relevance (semantic affinity in embedding space)—and IntuitiveGraphLLM, a framework that builds, encodes, and fuses IGs with pretrained LLMs. Given a passage, we first construct IGs by pruning structure-induced edges with a semantic gate based on cosine similarity between token (or span) embeddings, yielding sparse, human-plausible graphs. We then obtain initial node features from contextual embeddings and apply Graph Attention Networks (GATs) to emphasize informative nodes/edges to produce graph-level features. Finally, we perform hybrid fusion by integrating graph-level embeddings with LLM-based contextual representations, enabling the model to leverage complementary structural and conceptual signals. We evaluate our approach on five benchmark datasets spanning short and long documents and class-imbalance settings. Across benchmarks, IntuitiveGraphLLM consistently improves over strong text-only and graph-only baselines; gains persist under varied IG constructions, node embeddings, GAT depths/heads, and LLM backbones, with ablations confirming that IG is the key driver of performance and reduced edge noise. IntuitiveGraphLLM provides a principled, interpretable way to make text graphs both contextual and conceptually grounded, translating into more faithful reasoning and stronger downstream accuracy.

Introduction

In recent year, the advancement of Large Language Models (LLMs) has led to significant performance improvments across a wide range of NLP tasks and domains, including sentiment analysis (Cai et al. 2024; Rahman et al. 2024), biomedical retrieval (Xu et al. 2024), question answering (Robinson and Wingate 2023), code comprehension (Du et al. 2024), summarization and generation (Tu et al. 2024; He et al. 2024), and translation and text synthesis (Papi et al. 2024). The scaling data and model size have further expanded their capabilities (Wei et al. 2022b; Bubeck et al. 2023). Consequently, LLMs have attracted widespread attention from both academia (Wei et al. 2022a; Zhao et al. 2023) and industry (Achiam et al. 2023).

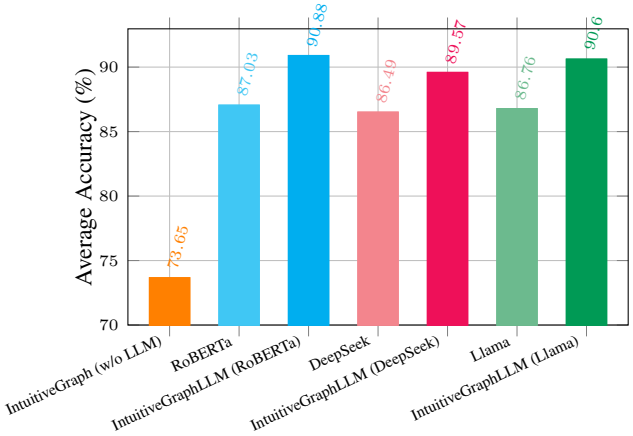
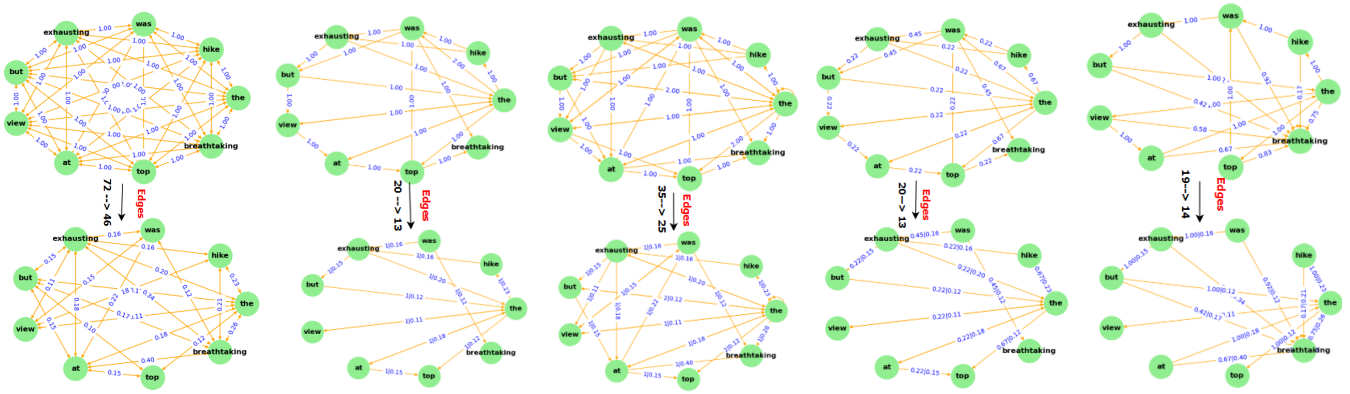


Figure 1: Average accuracy gains of IntuitiveGraphLLM variants (RoBERTa, DeepSeek, Llama) compared to their respective base LLMs and to IntuitiveGraph (w/o LLM), across datasets.

To adapt general-purpose LLMs for downstream tasks, numerous approaches have emerged. Beyond full-parameter fine-tuning, prompt- and prefix-based methods steer frozen models through learned prompts (Lester, Al-Rfou, and Constant 2021; Li and Liang 2021). Few-shot approaches enable an LLM to become a domain-specific model with limited examples (Brown et al. 2020). Parameter-efficient techniques that freeze pretrained weights and learn small rank-decomposition adapters reduce training cost while preserving performance (Tian et al. 2024). (Zhu et al. 2024; He et al. 2025) introduce novel parameter- and memory-efficient methods, such as ENGINE and UniGraph, which integrate LLMs with GNNs for textual graphs. Complementary directions integrate structured signals—e.g., knowledge graphs, hybrid feature pipelines, recurrent layers, and layer-specific adjustments—to enhance the structural and functional capacity of LLMs (Bugueño and de Melo 2023; Rahman et al. 2024). Despite the countless successes, LLMs often fail to retrieve and reason about the actual and complex semantics (or relationships) expressed in the text (Lewis et al. 2020; Pan et al. 2024). Graph-based representations



(a) Bag-of-Words (b) Window$_3$ (c) Window$_5$ (d) Sequence$_{weighted}$ (e) TextLevelGCN

Figure 2: Graph representations of the sentence *"The hike was exhausting, but the view at the top was breathtaking."* From left to right: (a) Bag-of-Words (BoW), (b) Window$_3$, (c) Window$_5$, (d) Sequence$_{weighted}$, and (e) TextLevelGCN. The top row uses structure-aware edges (co-occurrence/order/position). The bottom row shows the corresponding IGs.

offer a natural way to externalize and manipulate relational structure (Ji et al. 2021), but conventional text graphs, built from co-occurrence windows, sequential adjacency, or positional heuristics, can over-connect unrelated tokens and propagate noise. For example, in *"The hike was exhausting, but the view at the top was breathtaking,"* the conjunction *but* introduces a discourse contrast; capturing such structure-aware, conceptually meaningful relations is essential for faithful interpretation (Bugueño and de Melo 2023). Structure-aware graph encodings used in retrieval-augmented generation (Baek, Aji, and Saffari 2023; Lewis et al. 2020) may include extraneous noise, degrading LLM performance (Tian et al. 2024).

We propose IntuitiveGraphLLM, a framework that helps LLMs extract useful knowledge from Intuitive Graph (IG) representations of text. IGs first construct structure-driven graphs (e.g., windowed, sequential, position) and then apply a semantic gate, a cosine-similarity filter in embedding space, to retain edges that are both contextual and conceptually relevant. We initialize node features with domain-specific and contextual embeddings, and process IGs using Graph Attention Networks (GATs) (Veličković et al. 2018), which reweight neighborhoods to emphasize salient relations (e.g., the contrast signaled by *but*) while suppressing noise. In parallel, we encode the text with a pretrained LLM. The resulting hybrid representation—fusing graph-level and text-level features—combines explicit relational structure with rich contextualization.

To empirically evaluate our framework, we conducted extensive experiments on five benchmark datasets spanning biomedical and commonsense reasoning tasks. Results demonstrate that IntuitiveGraphLLM substantially enhances the semantic, structural, and logical understanding of text, producing consistent performance gains over both graph- and text-only baselines. As illustrated in Figure 1, IntuitiveGraphLLM with RoBERTa achieves an average accuracy improvement of **3.85%** (↑) compared to RoBERTa, while IntuitiveGraphLLM variants with DeepSeek and Llama improve by **3.08%** and **3.84%**, respectively, over their base models. IntuitiveGraphLLM not only surpasses the performance of backbone LLMs but also outperforms the structure-aware graph model without LLM integration (Bugueño and de Melo 2023) by **7.57%** (↑) on commonsense reasoning benchmarks, indicating the importance of fusing IG-based features with pretrained contextual representations. Moreover, for biomedical reasoning on PubMedQA, IntuitiveGraphLLM outperforms a recent state-of-the-art method (Tian et al. 2024) by a significant margin, highlighting IntuitiveGraphLLM's ability to generalize to domain-specific reasoning tasks. To summarize, our main contributions are:

- **Intuitive Graphs (IGs).** We formalize graphs (representation of text) that combine *structural context* (order/proximity/position) with *conceptual relevance* (semantic affinity), producing sparse, interpretable, and human-plausible structures.
- **IntuitiveGraphLLM.** We propose a framework that uses IGs with GATs for the graph branch and an LLM for the text branch; their embeddings (or features) are fused to improve robustness and precision on downstream tasks.
- **Comprehensive study.** We report results across five diverse datasets, multiple IG constructions and node-embedding choices, varied GAT settings, and LLMs, with ablations isolating the effects of semantic gating, graph processing, and fusion.

Background

Existing approaches to graphified text—such as BoW/TF-IDF graphs and sequential or positional schemes—typically connect tokens based on *structure alone* (co-occurrence, order, position) (Qian et al. 2024; Toroghi et al. 2024; Tian et al. 2024). These graphs often over-connect function words and adjacent tokens that are not conceptually related, injecting noise that weakens downstream reasoning with LLMs. We posit that graphs should retain edges only when they are both *contextually* and *conceptually* relevant, and we operationalize this via a *semantic gate* on top of standard structure-aware graph construction.

Figure 2 contrasts conventional graphs with their IG counterparts for the sentence *"The hike was exhausting, but*

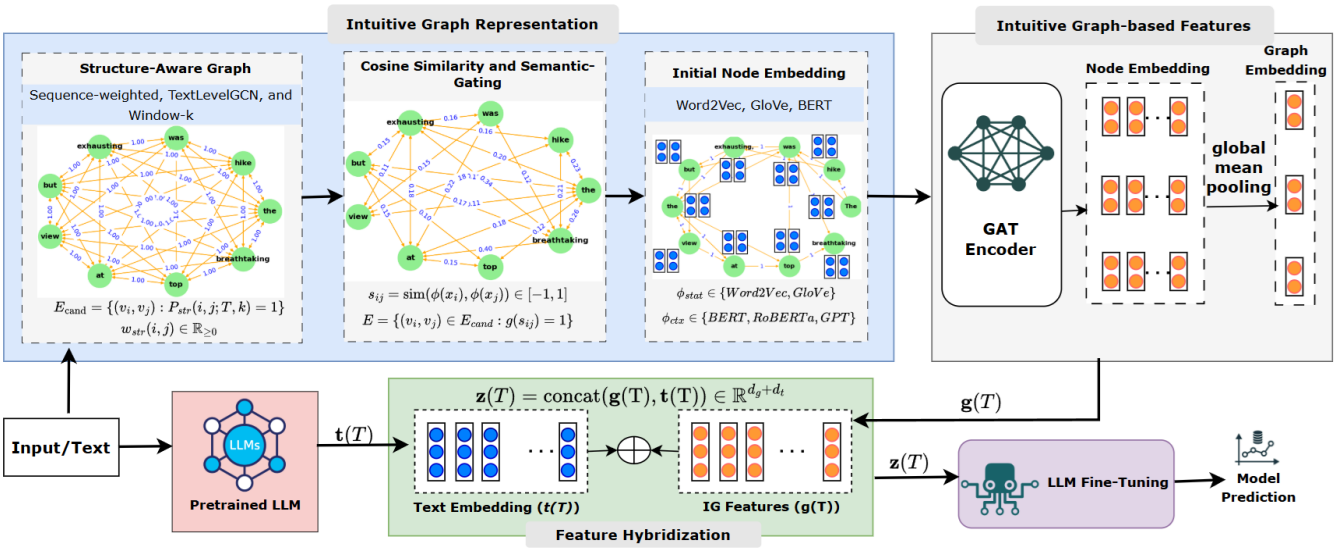


Figure 3: Overview of **IntuitiveGraphLLM**. Text is converted to an IG by forming structural candidate edges and pruning them via semantic-gating; nodes are initialized and encoded with a GAT to produce graph features ($g(T)$), which are fused with LLM-based contextual embeddings ($t(T)$) for end-to-end training and prediction.

the view at the top was breathtaking." All graphs are directed over the *nine unique tokens* in the sentence. IGs are derived by pruning structure-aware edges through a semantic gate, which retains only those edges whose token-embedding cosine similarity exceeds the threshold (e.g., threshold $\tau = 0.3$). Across five schemes, IGs reliably remove spurious links while keeping conceptually plausible edges: *BoW* drops from 72 to 46 edges (-36.1%); Window_3 from 20 to 13 (-35.0%); Window_5 from 35 to 25 (-28.6%); $\text{Sequence}_{weighted}$ from 20 to 13 (-35.0%); and TextLevelGCN from 19 to 14 (-26.3%). On average, IGs reduce edge count by $\approx 32\%$ across these families. Many removed edges involve weakly informative linkages among function or high-frequency words (e.g., `the` $\rightarrow$ `was`, `but` $\leftrightarrow$ `view`), which are structurally adjacent yet semantically unaligned.

Methodology

We decompose IntuitiveGraphLLM framework into four components, each formalized in the following subsections: (i) IG construction and initialization, (ii) IG processing with GATs and graph features, (iii) Text embedding with a pretrained LLM, and (iv) Feature fusion. Figure 3 summarizes the overall pipeline and information flow.

Constructing Intuitive Graph and Initializing Node Embeddings

The IG representation of text is designed to effectively capture the structural, contextual, and semantic relationships within an input sequence $T = \{x_1, x_2, \ldots, x_n\}$, where each x_i denotes a word (or token) in the sequence. Algorithm 1 is a pseudocode for IG construction and node feature initialization. An IG augments a structure-aware graph development concept with a semantic gate. Let E_{cand} be edges of a structural scheme (e.g., Window_k, BoW, $\text{Sequence}_{weighted}$, and TextLevelGCN). Edges are first built by a structural predicate $P_{\text{str}}(i, j; T, \kappa) \in \{0, 1\}$ controlled by a scheme κ:

$$E_{\text{cand}} = \{(v_i, v_j) \ : \ P_{\text{str}}(i, j; T, \kappa) = 1\}, \quad w_{\text{str}}(i, j) \in \mathbb{R}_{\geq 0}. \tag{1}$$

where w_{str} is a structural weight (count/TF–IDF/position). Let $\phi(\cdot) \in \mathbb{R}^{d_e}$ be an embedding function and $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$ be the cosine similarity. For each $(v_i, v_j) \in E_{\text{cand}}$ compute

$$s_{ij} \ = \ \text{sim}\big(\phi(x_i), \ \phi(x_j)\big) \in [-1, 1]. \tag{2}$$

Edges are retained by a policy g (e.g., threshold values)

$$E \ = \ \{(v_i, v_j) \in E_{\text{cand}} : g(s_{ij}) = 1\}. \tag{3}$$

Applying the semantic gating policy g removes structurally adjacent but semantically weak links (e.g., function-word ties), yielding sparser, more faithful neighborhoods. Next, we leveraged both static and contextual embedding methods to initialize the node embeddings. Let ϕ_{stat} be a static embedding (e.g., GloVe, Word2Vec) and $\phi_{\text{ctx}}(T)_i$ a contextual embedding for token x_i from a pretrained model (e.g., BERT). We form the initial node features $\mathbf{T}^{(0)} = [\mathbf{x}_1^{(0)}; \ldots; \mathbf{x}_n^{(0)}] \in \mathbb{R}^{n \times d}$ through static ($\mathbf{x}_i^{(0)} = \mathbf{P}\, \phi_{\text{stat}}(x_i)$) and contextual ($\mathbf{x}_i^{(0)} = \mathbf{P}\, \phi_{\text{ctx}}(T)_i$) embedding, where $\mathbf{P} \in \mathbb{R}^{d \times (\cdot)}$ is a learned projection.

Intuitive Graph Processing with GATs and Graph Features

Let the semantically gated IG be $G = (V, E, W)$ with $|V| = n$. From Section we obtain initial node embeddings $\mathbf{T}^{(0)} \in \mathbb{R}^{n \times d_{\text{in}}}$. The semantic gate fixes the (directed) neighborhood $\mathcal{N}(i) = \{\, j : (v_i, v_j) \in E \,\}$, and all attention is computed *only* over $\mathcal{N}(i)$ (i.e., pruned edges never receive attention weight). To maintain a clean analysis aligned with the IG

Algorithm 1: Constructing IGs and Initializing Node Embeddings

Input sequence $T = (x_1, x_2 \ldots, x_n)$; structural-aware graph scheme κ; embedding ϕ; static ϕ_{stat} and/or contextual ϕ_{ctx}; semantic threshold τ; projection $\mathbf{P}$. $\mathcal{IG} = (V, E, W)$, $\mathbf{T}^{(0)}$ $V \leftarrow \{v_1, \ldots, v_n\}$. Build structural candidates and weights: $E_{\text{cand}} \leftarrow \{(v_i, v_j) : P_{\text{str}}(i, j; T, \kappa) = 1\}$, $w_{\text{str}}(i, j) \in \mathbb{R}_{\geq 0}$. Precompute embeddings once:
$\mathbf{e}_i \leftarrow \phi(x_i) \in \mathbb{R}^{d_e}$ for $i = 1, \ldots, n$. For each $(v_i, v_j) \in E_{\text{cand}}$, compute cosine $s_{ij} \leftarrow \frac{\mathbf{e}_i^\top \mathbf{e}_j}{\|\mathbf{e}_i\| \|\mathbf{e}_j\|}$. Apply semantic gate (threshold):
$E \leftarrow \{(v_i, v_j) \in E_{\text{cand}} : s_{ij} \geq \tau\}$. $E =$ optional safeguard Keep $(v_p, v_q) = \arg\max_{(v_i, v_j) \in E_{\text{cand}}} s_{ij}$ $E \leftarrow \{(v_p, v_q)\}$. Set edge weights:
$W(i, j) \leftarrow w_{\text{str}}(i, j)$ for all $(v_i, v_j) \in E$. Initialize node embeddings:

$$\mathbf{x}_i^{(0)} \leftarrow \begin{cases} \mathbf{P}\, \phi_{\text{stat}}(x_i), \\ \mathbf{P}\, \phi_{\text{ctx}}(T)_i, \end{cases} \qquad i = 1, \ldots, n.$$

$(V, E, W), \mathbf{T}^{(0)} = [\mathbf{x}_1^{(0)}; \ldots; \mathbf{x}_n^{(0)}]$.

objective, this variant does not incorporate edge weights W into the attention mechanism. Instead, structural information is conveyed solely through the masked adjacency matrix E.

Layered GAT architecture and dimensions. We stack L graph-attention layers with hidden width d_h. Hidden layers use K heads with concatenation; the final layer uses one head without concatenation. Let $\mathbf{H}^{(\ell)} = [\mathbf{h}_1^{(\ell)}; \ldots; \mathbf{h}_n^{(\ell)}]$ denote node states at depth ℓ, with $\mathbf{H}^{(0)} = \mathbf{T}^{(0)}$. The dimensionality evolves as $\mathbf{H}^{(1)} \in \mathbb{R}^{n \times (K d_h)}$, $\mathbf{H}^{(\ell+1)} \in \mathbb{R}^{n \times (K d_h)}$ $(1 \leq \ell \leq L-2)$, $\mathbf{H}^{(L)} \in \mathbb{R}^{n \times d_{\text{out}}}$. A terminal linear projection $\mathbf{P} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{out}}}$ stabilizes the output:

$$\mathbf{Z} \;=\; \mathbf{H}^{(L)} \mathbf{P}^\top \in \mathbb{R}^{n \times d_{\text{out}}}. \tag{4}$$

Masked multi-head attention. Fix a layer ℓ and head m. Let $\mathbf{W}^{(\ell,m)}$ and $\mathbf{a}^{(\ell,m)}$ be learnable parameters. For $j \in \mathcal{N}(i)$, define attention logits and the masked softmax as

$$e_{ij}^{(\ell,m)} = \text{LeakyReLU}\Big(\mathbf{a}^{(\ell,m)\top} \Big[\, \mathbf{W}^{(\ell,m)} \mathbf{h}_i^{(\ell)} \;\Big\|\; \mathbf{W}^{(\ell,m)} \mathbf{h}_j^{(\ell)} \Big]\Big), \tag{5}$$

$$\alpha_{ij}^{(\ell,m)} = \frac{\exp\big(e_{ij}^{(\ell,m)}\big)}{\sum_{l \in \mathcal{N}(i)} \exp\big(e_{il}^{(\ell,m)}\big)}, \quad j \in \mathcal{N}(i). \tag{6}$$

Edges removed by the semantic gate do not appear in $\mathcal{N}(i)$ and therefore receive no attention mass.

Message passing, multi-head aggregation, and nonlinearity. With $\sigma = \text{ReLU}$, hidden layers aggregate per head and concatenate. The final layer aggregates with a single head (no concatenation):

$$\tilde{\mathbf{h}}_i^{(\ell+1)} = \Big\Vert_{m=1}^{K} \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(\ell,m)}\, \mathbf{W}^{(\ell,m)} \mathbf{h}_j^{(\ell)}, \tag{7}$$

$$\mathbf{h}_i^{(\ell+1)} = \sigma\big(\tilde{\mathbf{h}}_i^{(\ell+1)}\big), \qquad (0 \leq \ell \leq L-2), \tag{8}$$

$$\mathbf{h}_i^{(L)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(L-1,1)}\, \mathbf{W}^{(L-1,1)} \mathbf{h}_j^{(L-1)} \right), \tag{9}$$

$$\mathbf{H}^{(L)} = \big[\mathbf{h}_1^{(L)}; \ldots; \mathbf{h}_n^{(L)}\big]. \tag{10}$$

Equations 5–9 implement a GAT stack with multi-head hidden layers, single-head output, and ReLU after every layer—precisely the behavior of the module MULTILAYERGAT (hidden: K heads with concatenation; output: one head, no concatenation; final linear projection).

Graph embedding. Global mean pooling is applied to obtain the final node embeddings (or features), which are subsequently fused with the pretrained LLM representations.

$$\mathbf{g}(T) \;=\; \text{MeanPool}(\mathbf{Z}) \;=\; \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \;\in\; \mathbb{R}^{d_{\text{out}}}, \tag{11}$$

The IG mask reduces neighborhood size and degree variance, tightening the normalization in (6) and mitigating attention dilution from function-word ties. Multi-head hidden layers increase representational diversity on sparse IGs; the single-head output fixes the final width d_{out} for stable fusion. All structure flows through the masked adjacency E; weights W are not used inside attention in this variant, keeping the analysis clean and aligned with the IG objective.

Text Embedding with Pretrained LLM

In parallel to the IG branch, we obtain text representations from a pretrained LLM backbone. Given a tokenized input $T = (x_1, \ldots, x_n)$ with subword length M, the last hidden layer is $\mathbf{H}^{(L)} = \text{LLM}(T) \in \mathbb{R}^{M \times d_t}$. We extract a sequence embedding through a backbone-fit pooling function $p(\cdot)$:

$$\mathbf{t}(T) = p\Big(\mathbf{H}^{(L)}\Big), \quad p(\mathbf{H}) = \begin{cases} \mathbf{H}_0^{(L)} & \text{-encoder-only} \\ \frac{1}{|S|} \sum_{m \in S} \mathbf{H}_m^{(L)} & \text{-decoder-only} \end{cases}$$

$\mathbf{t}(T)$ is a compact, contextual summary of the entire sequence (e.g., the contrast signaled by "*but*" is encoded bidirectionally by the transformer). The text representation is then concatenated with the graph-level features in the fusion head (Section) and optimized end-to-end with the task loss.

Hybrid Feature Representation with IG- and LLM-based Features

In our proposed framework, a hybrid feature representation is constructed for the model. Let $\mathbf{g}(T) \in \mathbb{R}^{d_g}$ represent the *graph-level* features obtained from the IG branch (Section) and $\mathbf{t}(T) \in \mathbb{R}^{d_t}$ denote the *text-level* features generated by the LLM (Section). We build a hybrid feature representation through concatenation:

$$\mathbf{z}(T) \;=\; \text{concat}\big(\mathbf{g}(T),\, \mathbf{t}(T)\big) \;\in\; \mathbb{R}^{d_g + d_t}. \tag{12}$$

In our minimal implementation, we use the identity and fuse directly as in (12). A linear classifier maps the hybrid vector to logits for C classes:

$$\boldsymbol{\ell}(T) \;=\; \mathbf{W}\,\mathbf{z}(T) + \mathbf{b}, \qquad \mathbf{W} \in \mathbb{R}^{C\times(d_g+d_t)},\ \mathbf{b} \in \mathbb{R}^{C}. \tag{13}$$

Given a labeled dataset $\mathcal{D}_{\text{down}} = \{(T_j, y_j)\}_{j=1}^{M}$, we minimize the standard cross-entropy:

$$\mathcal{L}_{\text{down}} \;=\; -\frac{1}{M}\sum_{j=1}^{M} \log \frac{\exp(\ell_{y_j}(T_j))}{\sum_{c=1}^{C}\exp(\ell_c(T_j))}. \tag{14}$$

The above implementation corresponds to concatenating the text- and graph-level features, followed by an FC layer. A multi-layer MLP head (with nonlinearity, dropout, or normalization) is a drop-in generalization of (13). The fusion in (12) preserves *complementarity*: emphasizes *relational* cues distilled by the IG mask and GAT, whereas $\mathbf{t}(T)$ captures *global* context and lexical nuance from the LLM. Concatenation preserves both signals without requiring token-level alignment, keeps the head lightweight, and empirically stabilizes training when the two branches (graph and text) differ in scale or domain.

Experimental Settings

Datasets

Prior graphified text methods are often tailored to one task or to narrow text regimes (Yao, Mao, and Luo 2019), making robustness claims difficult to assess. Our selection stresses (i) *domain shift* (consumer, entertainment, general news, political news, biomedical), (ii) *document length* (short app reviews vs. long IMDB/HND articles), and (iii) *class balance*, to probe whether semantic gating in IGs consistently reduces structural noise and benefits downstream learning. To test whether IntuitiveGraphLLM transfers beyond a single domain or document style, we deliberately evaluate on five public corpora that vary in genre, length, and label skew, spanning sentiment, topic, ideology, and biomedical reasoning. Concretely, we use: App Reviews (Grano et al. 2017) (user reviews; sentiment), IMDB (Maas et al. 2011) (long-form movie reviews; sentiment), BBC News (Greene and Cunningham 2006) (news articles; topic classification), Hyperpartisan News Detection (HND) (Kiesel et al. 2018) (news articles), and PubMedQA (Jin et al. 2019) (biomedical question answering; we use the artificial subset and its yes/no labels as in our experiments). Table 1 summarizes their statistics, including average token length (ATL), class counts, and length distribution.

Dataset	Class	ATL	$\geq$ 100	$\geq$ 512	$\geq$ 1024
PubMedQA	2	27.60	0%	0%	0%
BBC News	5	459.34	100%	31.22%	2.27%
App Review	5	17.44	1.51%	0%	0%
IMDB	2	313.98	93.02%	14.92%	2.22 %
HND	2	1203.73	98.10%	72.93%	41.55%

Table 1: Summary statistics of the datasets. It reports the number of classes, ATL, and proportions of documents exceeding 100, 512, and 1024 tokens.

Model Settings

We configure IntuitiveGraphLLM by systematically varying its core components. (i) **Intuitive Graph methods.** We employ four IG constructions—Window$_3$, Window$_5$, Sequence$_{weighted}$, and TextLevelGCN. For TextLevelGCN, we vary the n-gram size from 1 to 3 to capture dependencies at different contextual ranges. During IG construction, we apply semantic gating with a cosine-similarity threshold $\tau \in [0.2, 0.5]$, ensuring that edges are retained only when both structurally relevant and semantically meaningful.

Estimating the Gating Threshold We estimate the semantic gate threshold τ for each dataset using Otsu's method (Otsu 1979) and the Benjamini–Hochberg FDR procedure (Benjamini and Hochberg 1995). For each document d, let V_d denote the token indices with contextual embeddings $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i\in V_d}$, and let $E^{(d)}_{\text{cand}}(K)$ be the set of structure-licensed candidate edges (e.g., *Window-K*). Edge cosine similarities are defined as in (2). To obtain a dataset-level operating point, we pool edge scores across a subset of documents. Specifically, for each $d \in \mathcal{D}_{\text{sub}}$, we compute $s^{(d)}_{ij}$ for all $(i,j) \in E^{(d)}_{\text{cand}}(K)$ and form $\mathcal{S} = \bigcup_{d\in\mathcal{D}_{\text{sub}}} \{\, s^{(d)}_{ij} \,\}$. We then estimate τ from $\mathcal{S}$ (via Otsu and FDR). Table 2 reports the resulting thresholds per dataset: the minimum is 0.4023 (Otsu, Window$_5$, BBC News) and the maximum is 1.00 (FDR, Window$_3$, PubMedQA). These estimates guide the choice of a global working threshold.

(ii) **Node embeddings.** Initial node representations are derived from three embedding schemes: BERT (768 dimensions), Word2Vec (300 dimensions), and GloVe (300 dimensions). (iii) **Graph encoder.** Graph embeddings are produced using a GAT with 2–4 hidden layers, each configured with 4 attention heads, and 128-dimensional hidden/output size. A *global mean pooling* layer aggregates node representations into graph-level embeddings. (iv) **LLM backbone.** For the text branch, we experiment with RoBERTa (`roberta-base`), DeepSeek (`DeepSeek-R1-Distill-Qwen-1.5b`), and LLaMA (`Llama-3.2-1B`) to provide contextual sequence-level representations. (v) **Optimization.** Models are trained with the Adam optimizer (learning rate 1×10^{-5}) and cross-entropy loss. These components—IG construction, node embeddings (NE), GAT encoders, and pretrained LLM backbones—are combined to form a unified hybrid framework for robust text representation learning.

Evaluation Metrics

We evaluate the performance of IntuitiveGraphLLM using standard classification metrics: *accuracy*, *precision*, *recall*, and *F1-score*, reported in both **macro-averaged** and **weighted-averaged** forms (Tian et al. 2024; Rahman et al. 2024).

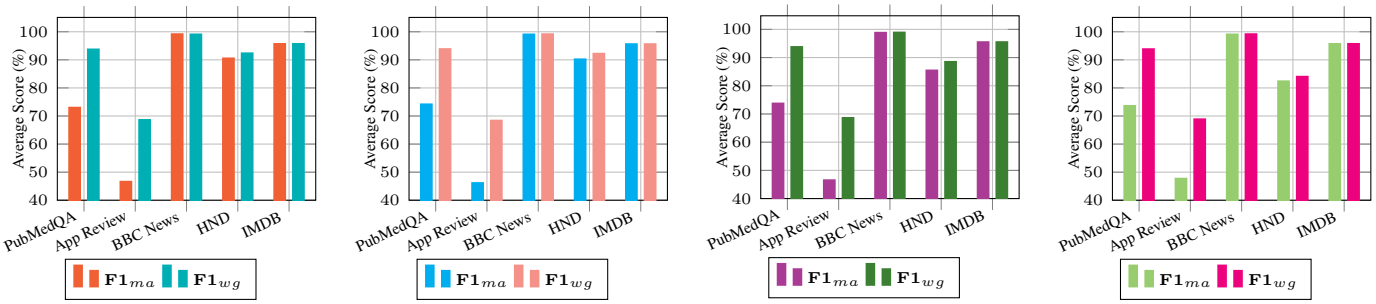
Performance Gain. To highlight the improvement of IntuitiveGraphLLM over its base LLM counterparts, we compute the relative gain:

$$\text{Gain}_{\Delta} = \text{Metric}_{\text{IntuitiveGraphLLM}} - \text{Metric}_{\text{Base LLM}}, \tag{15}$$

where Metric $\in \{\text{ACC}, \text{F1}_{ma}, \text{F1}_{wg}\}$. Positive $(+)$ values of Gain_{Δ} indicate a performance improvement $(\uparrow)$.

Dataset	Window$_3$			Window$_5$		
	$\lvert\mathcal{S}\rvert$	τ_{Otsu}	τ_{FDR}	$\lvert\mathcal{S}\rvert$	τ_{Otsu}	τ_{FDR}
HND	849,714	0.4258	0.5105	1,410,390	0.4180	0.4991
PubMedQA	111,096	0.4805	1.0000	177,160	0.4648	0.6622
BBC News	1,180,884	0.4258	0.5343	1,960,140	0.4023	0.5334
App Reviews	66,468	0.5273	-	103,730	0.5195	-
IMDB	970,866	0.4180	0.4675	1,610,110	0.4102	0.4862

Table 2: Dataset-level semantic gating threshold (τ) estimation under Window$_3$ and Window$_5$ structure-aware graphs. For each dataset, we report the pooled similarity count $|\mathcal{S}|$ and thresholds using Otsu and FDR approaches.



(a) Window$_3$ (b) Window$_5$ (c) Sequence$_{weighted}$ (d) TextLevelGCN

Figure 4: Comparison of the average $\mathbf{F1}_{ma}$ and $\mathbf{F1}_{wg}$ scores using different IG methods, GAT layers, and the RoBERTa LLM across all datasets.

Implementation Details

The experiments are conducted on a system running RHEL 8.8. The hardware configuration included an Intel Ice Lake (Xeon Platinum 8358) (2 sockets *32 cores/socket), 256 GB of RAM, and an NVIDIA A100 80GB PCIe graphics card.

Results

We evaluate the effectiveness of IntuitiveGraphLLM through extensive experiments on five benchmark datasets, systematically varying IG constructions, node embeddings, GAT depths, and LLM backbones. As a starting point, we focus on the RoBERTa-based variant, which consistently demonstrates strong performance across domains. IntuitiveGraphLLM (RoBERTa) achieves high Acc scores: 94.73 on PubMedQA, 72.39±0.5 on App Review, 99.19 on BBC News, 92.31 on HND, and 95.76 on IMDB. While these accuracy levels are competitive, performance on PubMedQA and App Review reveals comparatively lower macro $\mathbf{F1}_{ma}$ values, reflecting the effect of class imbalance. As shown in Figure 4, this trend persists across all datasets: macro $\mathbf{F1}_{ma}$ scores are consistently lower than weighted $\mathbf{F1}_{wg}$, underscoring the skewed class distributions.

Can IntuitiveGraphLLM Improve the Performance of Baseline LLMs?

We compared our best-performing IntuitiveGraphLLM variants against strong baselines, including BERT, RoBERTa, DeepSeek, and Llama (Table 3). Across all datasets, IntuitiveGraphLLM consistently matches or surpasses the performance of its corresponding backbone models. On PubMedQA, IntuitiveGraphLLM improves accuracy by +0.12%, +0.18%, and +0.77% over RoBERTa, DeepSeek, and Llama, respectively. Macro $\mathbf{F1}_{ma}$ scores also increase, with relative gains of +1.16% (RoBERTa), +1.55% (DeepSeek), and +4.97% (Llama). For the more challenging App Review dataset, where severe class imbalance suppresses $\mathbf{F1}_{ma}$, IntuitiveGraphLLM still provides consistent improvements. Notably, the IntuitiveGraphLLM variant (Llama) achieves gains of +3.39% in accuracy and +2.43% in $\mathbf{F1}_{ma}$ over Llama. On BBC News, which is relatively balanced, IntuitiveGraphLLM delivers substantial improvements. With RoBERTa, it yields +2.45% in accuracy and +2.48% in $\mathbf{F1}_{ma}$, while the Llama-based variant achieves a +1.63% accuracy gain. The most pronounced gains occur on the HND dataset, which contains long, complex political news articles. Here, IntuitiveGraphLLM improves accuracy by +10.78% to +12.31% and boosts $\mathbf{F1}_{ma}$ by +10.75% to +15.64%, depending on the backbone. On IMDB, IntuitiveGraphLLM again surpasses all baselines, improving accuracy by +5.48% over RoBERTa, +0.80% over DeepSeek, and +1.12% over Llama, with similar improvements in $\mathbf{F1}_{ma}$. These results demonstrate that IntuitiveGraphLLM not only enhances accuracy but also yields consistent improvements in macro $\mathbf{F1}_{ma}$ across diverse datasets. The largest benefits are observed on long-document and imbalanced datasets (HND, IMDB, BBC News), underscoring the role of IGs in mitigating noise and highlighting salient relations. The consistent gains across RoBERTa, DeepSeek, and Llama confirm that IG-augmented representation of text provide complementary inductive biases beyond what LLMs achieve alone.

Impact of the IG Features

To examine whether the observed gains arise merely from concatenation (*soft prompting*) rather than from meaningful

Model	PubMedQA		App Review		BBC News		HND		IMDB	
	Acc	$\mathbf{F1}_{ma}$	Acc	$\mathbf{F1}_{ma}$	Acc	$\mathbf{F1}_{ma}$	Acc	$\mathbf{F1}_{ma}$	Acc	$\mathbf{F1}_{ma}$
BoW MLP	90.74	65.23	67.65	41.69	95.12	94.93	78.46	72.12	87.60	87.60
BERT	94.14	72.31	71.20	44.84	96.74	96.57	73.84	68.86	88.80	88.79
RoBERTa	94.61	74.81	71.97	47.72	96.74	96.62	81.53	79.81	90.28	90.26
IntuitiveGraphLLM	94.73	73.65	72.39	46.17	99.19	99.10	92.31	90.56	95.76	95.76
Gain_{Δ}	↑ 0.12	↓ 1.16	↑ 0.42	↓ 1.55	↑ 2.45	↑ 2.48	↑ 10.78	↑ 10.75	↑ 5.48	↑ 5.50
DeepSeek	94.33	72.37	69.22	46.65	98.37	98.42	76.92	70.68	93.60	93.60
IntuitiveGraphLLM	94.51	73.92	70.52	46.92	99.19	99.24	89.23	86.32	94.40	94.40
Gain_{Δ}	↑ 0.18	↑ 1.55	↑ 1.30	↑ 0.27	↑ 0.82	↑ 0.82	↑ 12.31	↑ 15.64	↑ 0.80	↑ 0.80
Llama	93.75	69.81	69.09	44.37	97.56	97.61	80.00	76.84	93.40	93.40
IntuitiveGraphLLM	94.52	74.78	72.48	46.80	99.19	99.18	92.31	90.56	94.52	94.52
Gain_{Δ}	↑ 0.77	↑ 4.97	↑ 3.39	↑ 2.43	↑ 1.63	↑ 1.57	↑ 12.31	↑ 13.72	↑ 1.12	↑ 1.12

Table 3: Performance comparison of IntuitiveGraphLLM and state-of-the-art baseline models across the datasets.

graph information, we performed control experiments where the IG features/embeddings ($\mathbf{g}(T)$) were randomly permuted/shuffled before fusion with the LLM outputs ($\mathbf{t}(T)$). Table 4 reports the correlation metrics between the original and permuted IG embeddings across datasets. The near-zero values of cosine similarity (CS), mean Pearson (MP), and mean Spearman (MS) correlations confirm that the permutation completely disrupts the semantic structure; the permuted embeddings are orthogonal and uncorrelated with the originals. Figure 5 visualizes IG features for several IMDB samples before and after permuting/shuffling.

Dataset	Avg. CS	MP	MS
HND	0.0013	0.0012	0.0023
PubMedQA	0.005	-0.0091	0.0006
BBC News	0.0203	0.0146	0.0116
App Review	0.0128	0.0112	0.0076
IMDB	0.0161	0.0133	0.0106

Table 4: Correlation metrics before vs. after IG feature permuting/shuffling

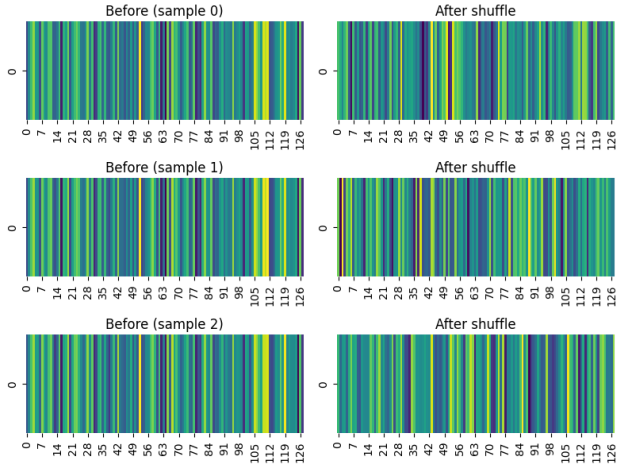


Figure 5: Visualization of IG feature representations for several IMDB samples before and after permuting/shuffling.

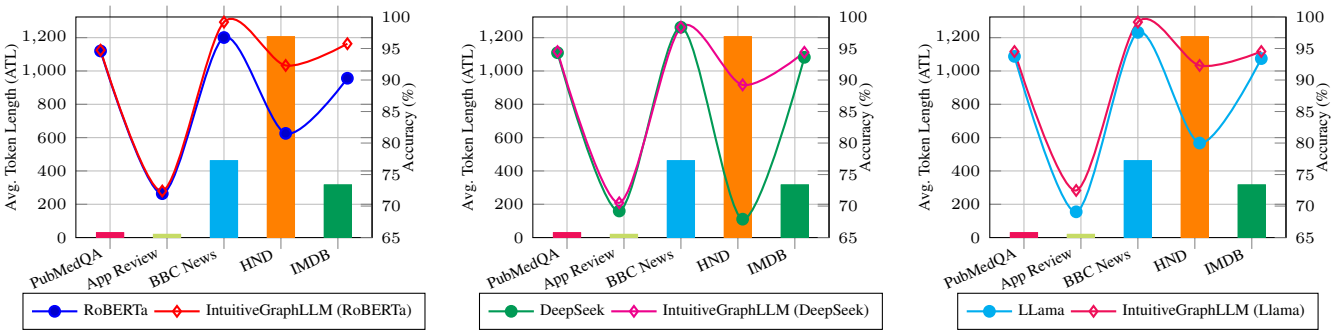
We then re-trained the IntuitiveGraphLLM (RoBERTa) model using these permuted features. As shown in Table 5, the Acc and $F1_{ma}$ scores dropped by approximately 11% on HND, 0.8% on IMDB, and 2.4% on BBC News. These declines indicate that aligned IG features carry non-trivial semantic signals that aid contextual understanding beyond text embeddings alone. For App Review and PubMedQA, the performance remained similar, likely because their ATL is short (see Table 1), limiting the graph's ability to encode additional structure. Consistent with our ablation results (see Table 6), IG contributes more strongly to long-document datasets, confirming that IntuitiveGraphLLM is especially more effective when semantic and relational dependencies extend across long tokens.

Model	Dataset	No Shuffle		After Shuffle	
		Acc	$F1_{ma}$	Acc	$F1_{ma}$
IntuitiveGraphLLM (RoBERTa)	HND	92.31	90.56	81.54↓	78.90↓
	IMDB	95.76	95.76	94.96↓	94.96↓
	BBC	99.19	99.10	96.75↓	96.84↓
	App Review	72.39	46.17	72.09↓	47.09↑
	PubMedQA	94.73	73.65	94.52↓	74.12↑

Table 5: Performance comparison before and after shuffling IG features.

Impact of Intuitive Graph Representation on LLM Performance

Table 1 summarizes the document length statistics of the five datasets. The HND dataset exhibits the longest documents, with an ATL of 1203.73. BBC News and IMDB also contain relatively long texts (ATLs of 459.34 and 313.98, respectively), whereas PubMedQA and App Review are considerably shorter. When examining length distributions, we find that 41.55% of HND articles exceed at least 1024 tokens, compared to only 2.27% of BBC News and 2.22% of IMDB documents, underscoring the particular challenge of modeling HND. However, to assess the effect of IGs on long-text modeling, we compared baseline LLMs with their IntuitiveGraphLLM counterparts. Figure 6 illustrates these results. With RoBERTa (Fig. 6a), IntuitiveGraphLLM achieves accuracy improvements of +2.45% on BBC News, +10.78% on HND, and +5.48% on IMDB. Using DeepSeek (Fig. 6b), the gain on HND is even larger at +12.31%. Similarly, with Llama (Fig. 6c), IntuitiveGraphLLM improves accuracy by +1.63% on BBC News, +12.31% on HND, and +1.12% on



(a) RoBERTa (b) DeepSeek (c) Llama

Figure 6: Comparison of performance gains between IntuitiveGraphLLM variants and their baseline LLM counterparts across datasets.

IMDB.

IntuitiveGraphLLM yields a significant average accuracy improvement on HND (+11.80%), IMDB (+2.47%), and BBC News (+1.36%). The obtained results highlight that performance gains are most pronounced on datasets with longer documents (e.g., HND), where structural noise and semantic drift pose greater challenges for baseline LLMs. Furthermore, consistent improvements on shorter-text datasets (e.g., PubMedQA, App Review) indicate that IG representations also benefit cases with limited context by strengthening structural and semantic alignment. Therefore, integrating IG into LLMs substantially enhances their ability to capture structural and semantic relations, with especially strong effects in long-document settings.

Ablation Study

The IntuitiveGraphLLM framework integrates two complementary components: (i) graph-based modules—IG construction, NE, and GATs—and (ii) a pretrained LLM for contextual representation. For this ablation study, we focused on RoBERTa and its IG-augmented variant. To quantify the contribution of each part, we performed ablation experiments by systematically removing components, with results reported in Table 6. When the LLM branch is removed (w/o LLM), performance drops sharply across datasets, particularly on App Review (68.23% Acc), BBC News (56.10% Acc), and HND (72.31% Acc), underscoring the importance of contextualized features towards the optimal model. Conversely, when the IG branch is removed (*w/o IG, NE, and GATs*), the LLM alone performs strongly on shorter and moderately long texts—for example, 96.74% on BBC News and 90.28% on IMDB. However, the full IntuitiveGraphLLM consistently attained the top results across all benchmarks. Accuracy improvements over the standalone LLM are +0.12% on PubMedQA, +0.42% on App Review, +2.45% on BBC News, +10.78% on HND, and +5.44% on IMDB. In particular, the gains are largest on long-document datasets such as HND and IMDB, where IG provide structural cues that complement LLM representations.

The ablation study findings highlight several key insights: (i) combining IG representations with LLM transforms the framework into a consistently top-performing model; (ii) semantic gating and graph-augmented processing benefit for capturing long-range dependencies and mitigating noise, particularly in complex and imbalanced datasets.

Dataset	Variant	Acc	$\mathbf{F1}_{ma}$	$\mathbf{F1}_{wg}$
PubMedQA	w/o LLM	94.34	66.88	92.73
	w/o IG	94.61	74.81	93.95
	IntuitiveGraphLLM	94.73	73.65	93.85
App Review	w/o LLM	68.23	33.72	61.10
	w/o IG	71.97	47.72	69.04
	IntuitiveGraphLLM	72.39	46.17	68.41
BBC News	w/o LLM	56.10	48.10	50.94
	w/o IG	96.74	96.62	96.76
	IntuitiveGraphLLM	99.19	99.10	99.19
HND	w/o LLM	72.31	41.96	60.69
	w/o IG	81.53	79.81	82.44
	IntuitiveGraphLLM	92.31	90.56	92.37
IMDB	w/o LLM	77.28	77.27	77.27
	w/o IG	90.28	90.26	90.28
	IntuitiveGraphLLM	95.72	95.71	95.72

Table 6: Performance of the models across datasets during ablation studies.

Conclusion

In this paper, we introduced IntuitiveGraphLLM, a hybrid framework that fuses semantically gated Intuitive Graphs with pretrained LLMs to strengthen structural, semantic, and logical understanding of text. By pruning noisy edges and preserving conceptually salient relations, IntuitiveGraphLLM provides interpretable graph structures that complement contextualized embeddings. Extensive experiments on five diverse benchmarks demonstrate consistent accuracy improvements across RoBERTa, LLaMA, and DeepSeek backbones, with average gains of 3.56% and up to 11.00% on long-document datasets such as HND. Comprehensive ablations further confirm that semantic gating and graph–LLM fusion are the key drivers of these improvements. The obtained results highlight IntuitiveGraphLLM as a principled and generalizable approach for enhancing LLMs in both short- and long-text reasoning tasks.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Baek, J.; Aji, A. F.; and Saffari, A. 2023. Knowledge-Augmented Language Model Prompting for Zero-Shot Knowledge Graph Question Answering. In Dalvi Mishra, B.; Durrett, G.; Jansen, P.; Neves Ribeiro, D.; and Wei, J., eds., *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, 78–106. Toronto, Canada: Association for Computational Linguistics.

Benjamini, Y.; and Hochberg, Y. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1): 289–300.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Bugueño, M.; and de Melo, G. 2023. Connecting the Dots: What Graph-Based Text Representations Work Best for Text Classification using Graph Neural Networks? In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8943–8960. Singapore: Association for Computational Linguistics.

Cai, H.; Ma, H.; Yu, J.; and Xia, R. 2024. A Joint Coreference-Aware Approach to Document-Level Target Sentiment Analysis. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12149–12160. Bangkok, Thailand: Association for Computational Linguistics.

Du, Y.; Ma, T.; Wu, L.; Zhang, X.; and Ji, S. 2024. AdaCCD: Adaptive Semantic Contrasts Discovery Based Cross Lingual Adaptation for Code Clone Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17942–17950.

Grano, G.; Di Sorbo, A.; Mercaldo, F.; Visaggio, C. A.; Canfora, G.; and Panichella, S. 2017. Android apps and user feedback: a dataset for software evolution and quality improvement. In *Proceedings of the 2nd ACM SIGSOFT international workshop on app market analytics*, 8–11.

Greene, D.; and Cunningham, P. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proceedings of the 23rd international conference on Machine learning*, 377–384.

He, J.; Yang, R.; Yu, L.; Li, C.; Jia, R.; Chen, F.; Jin, M.; and Lu, C.-T. 2024. Can We Trust the Performance Evaluation of Uncertainty Estimation Methods in Text Summarization? In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 16514–16575. Miami, Florida, USA: Association for Computational Linguistics.

He, Y.; Sui, Y.; He, X.; and Hooi, B. 2025. Unigraph: Learning a unified cross-domain foundation model for text-attributed graphs. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, 448–459.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2): 494–514.

Jin, Q.; Dhingra, B.; Liu, Z.; Cohen, W. W.; and Lu, X. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Kiesel, J.; Mestre, M.; Shukla, R.; Vincent, E.; Corney, D.; Adineh, P.; Stein, B.; and Potthast, M. 2018. Data for pan at semeval 2019 task 4: Hyperpartisan news detection. *(No Title)*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.-t.; Rocktäschel, T.; et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33: 9459–9474.

Li, X. L.; and Liang, P. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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)*, 4582–4597. Online: Association for Computational Linguistics.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In Lin, D.; Matsumoto, Y.; and Mihalcea, R., eds., *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1): 62–66.

Pan, S.; Luo, L.; Wang, Y.; Chen, C.; Wang, J.; and Wu, X. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering*.

Papi, S.; Gaido, M.; Negri, M.; and Bentivogli, L. 2024. StreamAtt: Direct Streaming Speech-to-Text Translation with Attention-based Audio History Selection. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the*

62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 3692–3707. Bangkok, Thailand: Association for Computational Linguistics.

Qian, X.; Zhang, Y.; Zhao, Y.; Zhou, B.; Sui, X.; Zhang, L.; and Song, K. 2024. TimeR^4 : Time-aware Retrieval-Augmented Large Language Models for Temporal Knowledge Graph Question Answering. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6942–6952. Miami, Florida, USA: Association for Computational Linguistics.

Rahman, M. M.; Shiplu, A. I.; Watanobe, Y.; and Alam, M. A. 2024. RoBERTa-BiLSTM: A Context-Aware Hybrid Model for Sentiment Analysis. *arXiv preprint arXiv:2406.00367*.

Robinson, J.; and Wingate, D. 2023. Leveraging Large Language Models for Multiple Choice Question Answering. In *The Eleventh International Conference on Learning Representations*. Kigali, Rwanda.

Tian, Y.; Song, H.; Wang, Z.; Wang, H.; Hu, Z.; Wang, F.; Chawla, N. V.; and Xu, P. 2024. Graph neural prompting with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 19080–19088.

Toroghi, A.; Guo, W.; Abdollah Pour, M. M.; and Sanner, S. 2024. Right for Right Reasons: Large Language Models for Verifiable Commonsense Knowledge Graph Question Answering. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6601–6633. Miami, Florida, USA: Association for Computational Linguistics.

Tu, L.; Yavuz, S.; Qu, J.; Xu, J.; Meng, R.; Xiong, C.; and Zhou, Y. 2024. Unlocking Anticipatory Text Generation: A Constrained Approach for Large Language Models Decoding. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 15532–15548. Miami, Florida, USA: Association for Computational Linguistics.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

Wei, J.; Bosma, M.; Zhao, V.; Guu, K.; Yu, A. W.; Lester, B.; Du, N.; Dai, A. M.; and Le, Q. V. 2022a. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022b. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Xu, R.; Shi, W.; Yu, Y.; Zhuang, Y.; Zhu, Y.; Wang, M. D.; Ho, J. C.; Zhang, C.; and Yang, C. 2024. BMRetriever: Tuning Large Language Models as Better Biomedical Text Retrievers. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 22234–22254. Miami, Florida, USA: Association for Computational Linguistics.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press. ISBN 978-1-57735-809-1.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Zhu, Y.; Wang, Y.; Shi, H.; and Tang, S. 2024. Efficient tuning and inference for large language models on textual graphs. *arXiv preprint arXiv:2401.15569*.