

# Internalizing Negation-Gated Logical Rules into LLMs for Document-Level Relation Extraction

Hye-Yoon Baek<sup>1</sup>, Sangjun Ji<sup>1</sup>, Jimyeung Seo<sup>1</sup>, Hae-Yoon Koo<sup>1</sup>,  
Xiongnan Jin<sup>2</sup>, Byungkook Oh<sup>1\*</sup>

<sup>1</sup>Konkuk University, Seoul, Republic of Korea

<sup>2</sup>Shenzhen University, Shenzhen, China

{hannah100, wltkdwns0220, rfyam8562, hykoo, bkoh}@konkuk.ac.kr  
xiongnanjin@szu.edu.cn

## Abstract

Document-level relation extraction (DocRE) requires integrating scattered evidence to uncover challenging implicit relations. While recent LLM-based approaches leverage rich prior knowledge for relation prediction, they suffer from two limitations. Pipeline-based approaches cause irrecoverable candidate pruning, and the absence of explicit logical constraints leaves spurious over-prediction unsuppressed. To address this, we propose Gated Logical-rule INTERNALization (GLINT), a novel DocRE framework that integrates an LLM with end-to-end inductive rule reasoning within a fully differentiable architecture. GLINT preserves per-pair signals across all entity pairs through a Fused Fact Tensor with Gumbel-softmax selection, and refines LLM predictions via an inductive rule reasoner with per-atom learnable negation gates. Extensive experiments on three benchmark datasets show that GLINT<sup>1</sup> outperforms existing LLM-based baselines.

## 1 Introduction

Document-level relation extraction (DocRE) aims to identify semantic relations between entity pairs within a document, supporting downstream tasks such as question answering and knowledge graph construction (Lee et al., 2026a; Ji et al., 2026). Unlike sentence-level relation extraction (Baek et al., 2025), which operates within a single sentence, DocRE requires resolving coreference among entity mentions and aggregating relational evidence across multiple sentences (Quirk and Poon, 2017; Yao et al., 2019). Traditional DocRE models often incorporate graph- or attention-based mechanisms to enrich semantic features of entity pairs (Zeng et al., 2020; Zhou et al., 2021). While effective at identifying relations explicitly presented in the

\*Corresponding author.

<sup>1</sup>Our code is available here: <https://github.com/hyeyoon100/GLINT>

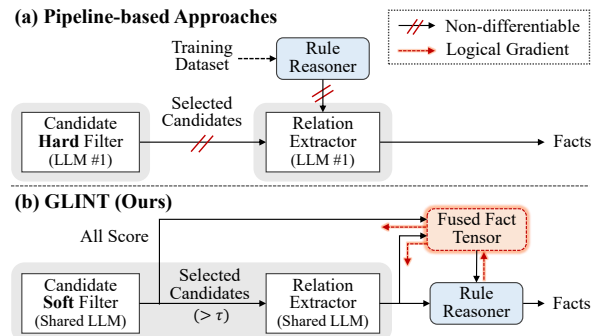


Figure 1: Comparison of (a) pipeline-based approaches and (b) our proposed GLINT for LLM-based DocRE. Pipelines train the candidate filter and relation extractor independently, causing irrecoverable entity-pair candidate pruning. GLINT jointly learns these modules and the rule reasoner within a single differentiable architecture, thereby internalizing negation-gated logical rules into the LLM. Red arrows indicate gradient flow.

text, they struggle with implicit relations that require multi-hop logical reasoning. To address this, recent approaches have introduced logical rules into DocRE frameworks, performing symbolic inference over explicit relations to uncover latent facts (Qi et al., 2024; Zhang et al., 2025b).

More recently, large language models (LLMs) have emerged as a promising direction for DocRE, leveraging their rich prior knowledge and strong contextual understanding (Gao et al., 2024; Xue et al., 2024). However, DocRE is particularly challenging for LLMs: documents are typically long, the label space is large, and most entity pairs hold no relation. For instance, in DocRED, over 97% of entity pairs are labeled NA (no relation), resulting in severe class imbalance. A common strategy in recent LLM-based DocRE methods is to adopt a two-stage pipeline (Figure 1): a candidate filtering module first prunes entity pairs predicted as NA, and a downstream module then performs relation classification over the retained pairs (Zhang et al., 2025a; Li et al., 2026).

A first limitation lies in the **irrecoverable can-**

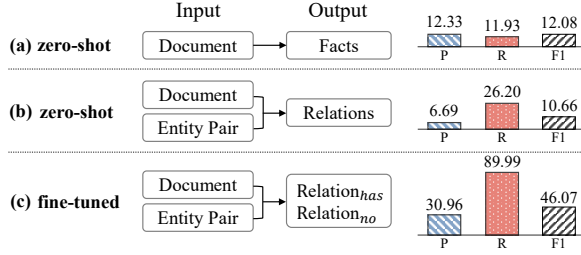


Figure 2: Precision, recall, and F1 of three LLM-based DocRE formulations on DocRED, varying in input granularity and supervision: (a) document-level zero-shot, (b) pair-level zero-shot, and (c) pair-level fine-tuned binary classification. Compared to (a), (b) reveals LLMs’ over-prediction tendency, dropping precision while raising recall. (c) improves both, yet recall continues to far exceed precision. See Appendix A for details.

**didate pruning** induced by this pipeline. The pipeline relies on discrete operations such as hard top- $k$  selection and token argmax, which break differentiability at every decision point. Once a candidate pair is pruned as NA, it is permanently excluded from subsequent reasoning. However, such pairs may hold implicit relations reachable only via multi-hop reasoning, or serve as essential intermediate nodes for deriving other implicit facts, providing signals the rule reasoner would otherwise rely on. With no continuous gradient path from the rule reasoner back to the upstream LLM backbone, the candidate filter and relation extractor are trained independently, limiting the model’s ability to refine earlier decisions based on later signals.

A second limitation lies in the **spurious over-prediction** of LLMs on DocRE. LLM-based methods can flexibly formulate fine-grained queries, e.g., at the single entity-pair level or by conditioning on a specific relation type (Xue et al., 2024; Zhang et al., 2025a). However, our preliminary experiments reveal that querying at the entity-pair level achieves high recall while sharply degrading precision, and fine-tuning only amplifies this imbalance (Figure 2). This reflects an inherent tendency of LLMs to draw on extensive prior knowledge extending beyond the literal text, yet existing methods provide no explicit mechanism to constrain the resulting spurious predictions.

Resolving these two limitations together requires equipping the model to recover missing facts through multi-hop reasoning while suppressing spurious ones through explicit logical constraints. This motivates **internalizing logical rules into the LLM itself**. Recent work has highlighted the potential of shaping LLM internal representations to improve knowledge integration and rea-

soning (Jeong et al., 2025). In GLINT, the rule reasoner doubles as both an expressive inference engine that *constrains the LLM’s spurious over-predictions* through positive, negative, and mixed conditions, and the bridge through which logical gradients propagate back to refine the candidate filter and relation extractor, thereby *recovering pairs that pipelines would otherwise prune irrecoverably*. Realizing this design poses two non-trivial challenges: (1) maintaining a differentiable end-to-end pathway across an inherently discrete LLM while keeping query cost sub-quadratic over all  $N^2$  entity pairs, and (2) incorporating negative conditions into a unified rule module without introducing conflicting gradients between sufficient- and necessary-condition objectives.

To address these challenges, we propose **Gated Logical-rule INTERNALization (GLINT)**, a novel LLM-based DocRE model that jointly learns entity-pair filtering, relation extraction, and negation-gated rule reasoning in an end-to-end manner. GLINT couples three modules through the *Fused Fact Tensor*  $\mathbf{F}^+$  which fuses and softly preserves the per-pair relation signals across all entity pairs: (1) a soft candidate filter that retains all entity-pair signals including those filtered as NA via  $\mathbf{F}^+$  preservation and Gumbel-softmax selection, (2) an LLM-based relation extractor, and (3) an inductive rule reasoner that unifies positive, negative and mixed conditions through per-atom learnable negation gates. The rule reasoner refines LLM predictions through chain reasoning over the preserved signals at inference, while its logical gradients flow back through  $\mathbf{F}^+$  to refine the filter and extractor at every training step, enabling continuous signal exchange across the entire model. Our main contributions are summarized as follows:

- To the best of our knowledge, GLINT is the first DocRE model to integrate an LLM backbone with end-to-end inductive rule learning.
- We replace hard top- $k$  filtering with a fully differentiable selection that preserves logit signals from every entity pair.
- We extend soft rule learning to both positive and negative directions via per-atom learnable negation gates.
- Extensive experiments on DocRED, Re-DocRED, and DWIE show that GLINT outperforms strong baselines, with particularly large gains on implicit relations.

## 2 Related Work

### 2.1 Document-level Relation Extraction

Existing approaches enhance entity-pair representations through either graph structures or attention mechanisms. LSR (Nan et al., 2020) learns a latent document structure to facilitate multi-hop reasoning, and GAIN (Zeng et al., 2020) strengthens document-level reasoning by jointly leveraging mention-level and entity-level graphs. ATLOP (Zhou et al., 2021) introduces adaptive thresholding together with localized context pooling for multi-label relation prediction, while DREEM (Ma et al., 2023a) exploits evidence information to guide model attention toward relation-supporting sentences.

While effective at encoding document-level contextual information, these methods do not explicitly capture logical dependencies among extracted relations. As a result, their predictions are not constrained by document-level logical consistency.

### 2.2 Logic Rule-based DocRE

Since relation facts in a document often exhibit logical dependencies (e.g., implication, transitivity, inverse relations), several studies have incorporated logic rules into DocRE to improve consistency and reasoning. Earlier methods treat rule learning as a separate component, where LogiRE (Ru et al., 2021) treats logic rules as latent variables under EM-style optimization, MILR (Fan et al., 2022) injects mined rules via consistency regularization, and BCBR (Liu et al., 2023) mines rules with a beta-distribution-based miner with forward and reverse constraints to suppress spurious rules. Departing from this two-stage view, JMRL (Qi et al., 2024) jointly optimizes relation extraction and rule learning end-to-end, while CaDRL (Zhang et al., 2025b) learns document-specific rules.

However, these approaches remain confined to PLM-based DocRE backbones and to rule formulations using only positive conditions. As a result, both the integration of logic rules with LLMs and the use of negative or mixed conditions to constrain spurious predictions remain largely underexplored.

### 2.3 LLM-based DocRE

Recent studies have applied LLMs to DocRE but reported that direct in-context learning struggles with fine-grained relation types and hallucinated outputs (Li et al., 2023), motivating a shift toward selecting candidate entity pairs or relations

before extraction. AutoRE (Xue et al., 2024) introduces the RHF (Relation-Head-Facts) paradigm that sequentially predicts relations, heads, and facts with QLoRA fine-tuning. LMRC (Li et al., 2026) pre-selects candidate pairs via a relation classifier to mitigate attention distraction. EP-RSR (Zhang et al., 2025a) proposes the EPRF (Entity Pair-Relation-Fact) paradigm, which retrieves relation-relevant context per entity pair before extraction. However, these approaches adopt two-stage pipelines in which the candidate selector and the relation extractor are trained with separate objectives, and none of them explicitly models logical dependencies among relation facts. RuAG (Zhang et al., 2025c) injects logic rules into LLM prompts as natural language via MCTS-based rule learning, but this prompt-level integration keeps rule learning and LLM extraction decoupled, preventing end-to-end joint optimization.

Whereas existing LLM-based methods decouple candidate selection from relation extraction, GLINT pursues a tighter integration by learning them jointly within a single end-to-end framework.

## 3 Preliminaries

**Logical Rule.** An atom  $p(x, y)$ , where  $p$  is a predicate and  $x, y$  are entity variables, is a binary literal indicating whether predicate  $p$  holds between  $x$  and  $y$ . Predicates are distinguished by their role: head predicates, denoted by  $r$ , are drawn from  $\mathcal{R}$ , while body predicates, denoted by  $B$ , are drawn from the extended set  $\mathcal{R}^+ = \mathcal{R} \cup \mathcal{R}^- \cup \{I\}$  of size  $2n + 1$ , where  $\mathcal{R}^- = \{r^- \mid r \in \mathcal{R}\}$  contains inverse predicates and  $I$  is the identity predicate. We adopt chain-like logical rules (CRs) (Abiteboul et al., 1995), in which any two consecutive body atoms share an intermediate variable. An  $L$ -CR with head predicate  $r$  takes the form:

$$r(x, y) \leftarrow B_1(x, z_1) \wedge B_2(z_1, z_2) \wedge \cdots \wedge B_L(z_{L-1}, y),$$

where  $B_1, \dots, B_L \in \mathcal{R}^+$  are body predicates, and  $z_1, \dots, z_{L-1}$  are intermediate entity variables.

**Problem Formulation.** Given a document  $d \in \mathcal{D}$  and its set of named entities  $\mathcal{E}_d = \{e_i\}_{i=1}^{n_d}$ , where each entity  $e_i$  may have multiple mentions throughout  $d$ , DocRE aims to identify the relational triples  $(e_h, r, e_t)$  that hold in  $d$ , where  $e_h, e_t \in \mathcal{E}_d$ ,  $e_h \neq e_t$ , and  $r \in \mathcal{R}$  is one of  $n = |\mathcal{R}|$  predefined relations. We frame DocRE as estimating, for each predicate  $r \in \mathcal{R}$  and each entity pair  $(e_h, e_t)$ , a

soft score  $\phi_r(e_h, e_t) \in [0, 1]$  that represents the estimated truth value of the atom  $r(e_h, e_t)$ . We learn this score by fine-tuning a low-rank adapter (LoRA) on the LLM jointly with a neuro-symbolic rule module.

## 4 Methodology

As illustrated in Figure 3, GLINT consists of a Differentiable Candidate-pair Filter (DCF) and a Generative Relation Extractor (GRE) sharing a single fine-tuned LLM backbone, together with a Rule Reasoner. The Fused Fact Tensor  $\mathbf{F}^+$  aggregates their per-pair outputs and feeds the Rule Reasoner, whose logical gradients flow back through to refine both modules. The model is trained end-to-end by minimizing a weighted sum of classification losses.

### 4.1 Differentiable Candidate-Pair Filter

Initially, we determine whether *any* relation holds between an entity pair, rather than identifying *which* relations exist. By pruning the search space from  $O(n_d^2)$  to a small candidate set  $\mathcal{C}$ , this module lets the subsequent stage attend only to promising pairs. As the entry stage, it upper-bounds the recall of the entire framework. We therefore design it to be lightweight while prioritizing recall.

**Entity and relation representations.** The LLM takes as input a prefix  $(I, d, \mathcal{E}_d)$ , where  $I$  is a task instruction. We append a special token [ENT] after each entity query  $e_i : \text{name}_i$  and run a single forward pass under a block-diagonal attention mask. The prefix uses standard causal attention, while each entity query attends only to the prefix and itself. The resulting [ENT] hidden state gives the entity representation  $\mathbf{e}_i \in \mathbb{R}^{d_{\text{LLM}}}$ . By attending to the document  $d$  through the prefix,  $\mathbf{e}_i$  is document-dependent. In contrast, the representation  $\mathbf{r}_k \in \mathbb{R}^{d_{\text{LLM}}}$  of relation  $r_k \in \mathcal{R}$  is built without any document context. We tokenize its natural-language label with the LLM tokenizer and mean-pool the input embeddings of its subword tokens. This grounds each relation in the LLM’s pretrained semantics, making it document-agnostic.

**Triple scoring and selection.** To score the plausibility of relation  $r_k$  holding between  $e_h$  and  $e_t$ , we introduce a *Fact Scorer*, based on a knowledge graph completion (KGC) scorer. Among standard KGC methods, we adopt DistMult (Yang et al., 2015) for its low computational cost and sufficient expressiveness in filtering; further analysis is provided in Appendix F.1. DistMult is symmet-

ric, whereas DocRE relations are directional. We therefore apply separate head- and tail-specific projections  $\hat{\mathbf{e}}_h = \mathbf{W}_h \mathbf{e}_h$  and  $\hat{\mathbf{e}}_t = \mathbf{W}_t \mathbf{e}_t$ , together with a relation projection  $\hat{\mathbf{r}}_k = \mathbf{W}_r \mathbf{r}_k$ , where  $\mathbf{W}_h, \mathbf{W}_t, \mathbf{W}_r \in \mathbb{R}^{d_p \times d_{\text{LLM}}}$ . The relation score is

$$z_{h,t,k}^{\text{DCF}} = \hat{\mathbf{e}}_h^\top \text{diag}(\hat{\mathbf{r}}_k) \hat{\mathbf{e}}_t, \quad (1)$$

which gives  $z_{h,t,k}^{\text{DCF}} \neq z_{t,h,k}^{\text{DCF}}$  in general. Then we aggregate the per-relation scores into a single pair-level score,

$$s_{h,t} = \log \frac{1}{n} \sum_{k=1}^n \exp(z_{h,t,k}^{\text{DCF}}), \quad (2)$$

which captures whether *any* relation likely holds for the pair. We use log-mean-exp pooling rather than a hard maximum so that  $s_{h,t}$  remains differentiable.

To form the candidate set  $\mathcal{C}$ , we relax the per-pair binary selection decision with a binary Gumbel-Softmax (Jang et al., 2016; Maddison et al., 2017),

$$\tilde{g}_{h,t} = \sigma\left(\frac{s_{h,t} + L_{h,t}}{\tau}\right), \quad (3)$$

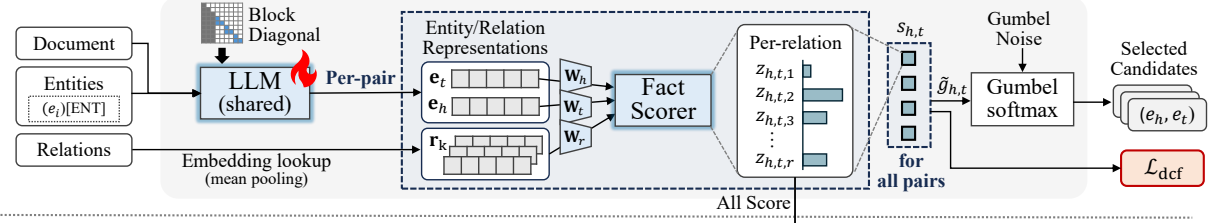
where  $L_{h,t} \stackrel{\text{i.i.d.}}{\sim} \text{Logistic}(0, 1)$  and  $\tau$  is a temperature parameter. This relaxation makes selection differentiable, allowing gradients to flow through  $\tilde{g}_{h,t}$  during training. At inference,  $\tilde{g}_{h,t}$  reduces to the deterministic  $\sigma(s_{h,t}/\tau)$ . The candidate set  $\mathcal{C}$  is then formed by retaining pairs with  $\tilde{g}_{h,t} > \theta$ , capped at the top- $K$  per document. Crucially, the full triple-level tensor  $\mathbf{z}^{\text{DCF}}$  is not discarded but preserved and passed to the rule reasoning stage.

### 4.2 Generative Relation Extractor

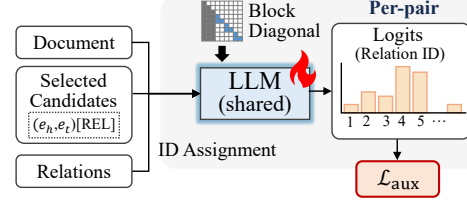
Following the candidate entity pair filtering, we exploit the LLM’s generative capacity to predict *which* relations each candidate pair holds. The LLM receives the prefix  $(I, d, \mathcal{E}_d, \mathcal{R})$ , and each relation is assigned a unique reserved token in the LLM vocabulary. Relation prediction thus reduces to next-token prediction. We read the next-token logits over the relation tokens as independent per-relation scores. As  $\mathcal{C}$  may still be large, we apply the same sequence packing as in DCF: each pair query is followed by a [REL] token and packed under a block-diagonal mask, so that each query attends only to the shared prefix and itself. This amortizes prefix encoding while preventing cross-pair interference (see Algorithm 1 in Appendix B for details).

For each pair  $(e_h, e_t) \in \mathcal{C}$ , the output hidden state at the [REL] position  $\mathbf{h}_{h,t}^{[\text{REL}]} \in \mathbb{R}^{d_{\text{LLM}}}$  is

### (1) Differentiable Candidate-Pair Filter



### (2) Generative Relation Extractor



### (3) Rule Reasoner

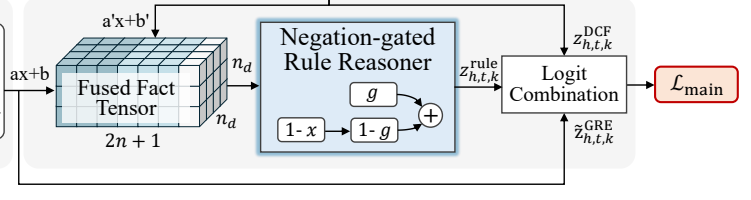


Figure 3: The overview of our proposed model GLINT for LLM-based DocRE. GLINT jointly learns (1) Differentiable Candidate-pair Filter, (2) Generative Relation Extractor, and (3) Rule Reasoner within a single differentiable architecture through the Fused Fact Tensor  $\mathbf{F}^+$ , thereby internalizing negation-gated logical rules into the LLM.

passed through the LM head and restricted to the relation token IDs  $\mathcal{I}_{\mathcal{R}}$ , yielding the relation logit vector  $\mathbf{z}_{h,t}^{\text{GRE}} \in \mathbb{R}^n$ :

$$\mathbf{z}_{h,t}^{\text{GRE}} = \text{LMHead}(\mathbf{h}_{h,t}^{\text{REL}})[\mathcal{I}_{\mathcal{R}}]. \quad (4)$$

These scores are passed to the rule reasoning stage.

### 4.3 Rule Reasoner over Fused Fact Tensor

**$\mathbf{F}^+$  construction.** We construct the *fused fact tensor*  $\mathbf{F}^+$ , which stores the predicted soft probability of each candidate triple by fusing the complementary outputs of DCF and GRE. DCF produces a dense triple-level tensor  $\mathbf{z}^{\text{DCF}} \in \mathbb{R}^{n_d \times n_d \times n}$  over all pairs, while GRE produces  $\mathbf{z}^{\text{GRE}}$  only on the candidate set  $\mathcal{C}$ . The two are complementary:  $\mathbf{z}^{\text{DCF}}$  covers all pairs with coarse per-relation scores, and  $\mathbf{z}^{\text{GRE}}$  covers  $\mathcal{C}$  only but with sharper per-relation predictions. We fuse the two to retain both.

As the two signals lie on different scales, we calibrate each with a learnable affine map followed by a sigmoid. For  $\alpha \in \{\text{DCF}, \text{GRE}\}$ :

$$\hat{p}_{h,t,k}^\alpha = \sigma(a_\alpha \mathbf{z}_{h,t,k}^\alpha + b_\alpha), \quad (5)$$

where the four scalars ( $a_{\text{DCF}}, b_{\text{DCF}}, a_{\text{GRE}}, b_{\text{GRE}}$ ) are initialized to the identity map ( $a = 1, b = 0$ ). The calibrated logit  $\tilde{z}_{h,t,k}^{\text{DCF}}$  is also reused (without sigmoid) as the non-candidate term in the rule-augmented logit  $\phi$  (Eq. 13). The hard mask  $m_{h,t} = \mathbb{1}[(h, t) \in \mathcal{C}]$  from thresholding  $\tilde{g}_{h,t}$  is non-differentiable, so we restore the gradient with a straight-through estimator,

$$\hat{m}_{h,t} = m_{h,t} + \tilde{g}_{h,t} - \text{SG}(\tilde{g}_{h,t}), \quad (6)$$

where  $\text{SG}(\cdot)$  denotes the stop-gradient.  $\hat{m}_{h,t}$  equals  $m_{h,t}$  in the forward pass but admits a non-trivial gradient through  $\tilde{g}_{h,t}$  in the backward pass. Each forward cell mixes two calibrated sources via  $\hat{m}_{h,t}$ :

$$f_{h,t,k} = \hat{m}_{h,t} \hat{p}_{h,t,k}^{\text{GRE}} + (1 - \hat{m}_{h,t}) \hat{p}_{h,t,k}^{\text{DCF}}, \quad (7)$$

with diagonal cells ( $h = t$ ) zeroed. Candidate pairs, indicated by  $\hat{m}_{h,t} = 1$ , take the GRE prediction, while non-candidate pairs fall back to the DCF prediction. The fused fact tensor  $\mathbf{F}^+$  then materializes per-pair scores over the extended predicate set  $\mathcal{R}^+ = \mathcal{R} \cup \mathcal{R}^- \cup \{I\}$ :

$$\mathbf{F}^+(h, t, \cdot) = [\mathbf{f}_{h,t} \parallel \mathbf{f}_{t,h} \parallel \mathbb{1}[h=t]] \in \mathbb{R}^{2n+1}, \quad (8)$$

where the inverse block  $\mathbf{f}_{t,h}$  is obtained by transposition and the identity indicator marks self-pairs. The resulting  $\mathbf{F}^+ \in [0, 1]^{n_d \times n_d \times (2n+1)}$  feeds the rule-reasoning module.

**Negation-gated rule reasoning.** We extend JMRL’s chain-style rule learner (Qi et al., 2024) to operate over  $\mathbf{F}^+$ , yielding a rule-based logit  $\mathbf{z}^{\text{rule}}$  added as a residual to the final prediction logits. A chain rule of depth  $L$  composes  $L$  literals along an entity path  $x_0 \rightarrow \dots \rightarrow x_L$  as a soft conjunction, with each literal reading from  $\mathbf{F}^+$  via a learnable softmax over the  $2n + 1$  predicates. For each relation  $r_k$ , we instantiate  $N$  such rules.

Our extension is a *per-atom negation gate* that lets a rule select negated literals, broadening the rule class from positive-only conjunctions to patterns involving  $\neg p$ . For relation  $r_k$ , rule  $j$ , and hop  $\ell$ , the pre-gate atom  $\tilde{a}_{k,j}^{(\ell)}(x, y) \in [0, 1]$  is the softmax-weighted read of  $\mathbf{F}^+(x, y, \cdot)$  over the

$2n + 1$  predicates. We attach a sigmoid gate  $g_{k,j}^{(\ell)} = \sigma(v_{k,j}^{(\ell)}) \in (0, 1)$  on a learnable scalar  $v_{k,j}^{(\ell)}$ , forming the literal,

$$a_{k,j}^{(\ell)}(x, y) = g_{k,j}^{(\ell)} \tilde{a}_{k,j}^{(\ell)}(x, y) + (1 - g_{k,j}^{(\ell)})(1 - \tilde{a}_{k,j}^{(\ell)}(x, y)), \quad (9)$$

which equals  $\tilde{a}$  at  $g \rightarrow 1$  (positive predicate) and  $1 - \tilde{a}$  at  $g \rightarrow 0$  (its negation). We initialize  $v$  to 0 so that every literal starts polarity-neutral and learns its sign from data.

The  $L$  literals are composed into a soft relational chain over intermediate entities and aggregated across the  $N$  rules with signed tanh confidences, yielding  $z_{h,t,k}^{\text{rule}}$  for each pair  $(h, t)$  and relation  $r_k$ ; the full composition and aggregation formulas appear in Appendix C. The two mechanisms play complementary roles: the negation gate flips atom polarity within  $[0, 1]$  and thus keeps per-literal contributions non-negative while the signed tanh confidences in aggregation provide the sign of each rule’s contribution to  $z^{\text{rule}}$ . Together they express four rule forms by composing positive or negated literals with positive or negative confidence, covering rule patterns with negated atoms that prior soft rule learners with positive-only formulations cannot capture.

#### 4.4 Training Objective

We train GLINT end-to-end with three binary cross-entropy losses:

$$\mathcal{L}_{\text{main}} = \text{BCE}(\sigma(\phi_{h,t,k}), y_{h,t,k}), \quad (10)$$

$$\mathcal{L}_{\text{aux}} = \text{BCE}(\sigma(z_{h,t,k}^{\text{GRE}}), y_{h,t,k}), \quad (11)$$

$$\mathcal{L}_{\text{dcf}} = \text{BCE}(\sigma(s_{h,t}), y_{h,t}^{\text{pair}}), \quad (12)$$

where  $y_{h,t,k}$  indicates whether relation  $r_k$  holds between  $e_h$  and  $e_t$ , and  $y_{h,t}^{\text{pair}}$  is positive if the pair holds at least one relation.  $\mathcal{L}_{\text{dcf}}$  provides pair-level supervision to the candidate filter,  $\mathcal{L}_{\text{aux}}$  provides relation-level supervision to GRE on the selected candidate pairs, and  $\mathcal{L}_{\text{main}}$  supervises the final prediction for each relation over all entity pairs, including those not selected as candidates. The main loss is computed from the full rule-augmented prediction logit

$$\phi_{h,t,k} = \hat{m}_{h,t} z_{h,t,k}^{\text{GRE}} + (1 - \hat{m}_{h,t}) \tilde{z}_{h,t,k}^{\text{DCF}} + z_{h,t,k}^{\text{rule}}. \quad (13)$$

The first two terms form a selection-dependent backbone prediction. Selected candidate pairs

use the fine-grained GRE logits, whereas non-candidate pairs retain the calibrated DCF logits instead of being discarded. The rule logit is added as a signed residual, allowing the rule reasoner to increase or decrease the final relation score.

The three losses provide complementary gradient paths to the shared LLM backbone.  $\mathcal{L}_{\text{dcf}}$  updates DCF and the backbone through the entity-representation forward pass, while  $\mathcal{L}_{\text{aux}}$  updates GRE and the backbone through the relation-extraction forward pass. Within  $\mathcal{L}_{\text{main}}$ , the direct  $z_{h,t,k}^{\text{GRE}}$  term remains connected to the computation graph, allowing gradients from the rule-augmented prediction loss to update GRE and the shared backbone on selected pairs. The straight-through mask  $\hat{m}_{h,t}$ , which uses hard selection in the forward pass and the soft Gumbel selection as a backward surrogate, additionally routes gradients through the candidate-selection decision to DCF and the backbone. Stop-gradient is applied to the raw DCF logit in the non-candidate branch of  $\phi$  and to the raw DCF and GRE logits used to construct  $\mathbf{F}^+$ , blocking backpropagation through these branches while preserving the direct GRE path in  $\phi$  and the straight-through selection path through  $\hat{m}_{h,t}$ . Consequently,  $\mathcal{L}_{\text{main}}$  jointly trains GRE, DCF, the calibration parameters, and the rule reasoner.

The full objective is:

$$\mathcal{L} = \lambda_{\text{dcf}} \mathcal{L}_{\text{dcf}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}} + \mathcal{L}_{\text{main}}, \quad (14)$$

where  $\lambda_{\text{dcf}}$  and  $\lambda_{\text{aux}}$  balance the two auxiliary losses. Because DCF and GRE share the same LoRA parameters, these complementary gradient paths jointly optimize a single LLM backbone.

## 5 Experiments and Analysis

In this section, we present a comprehensive empirical evaluation of our proposed model, **GLINT**. Our experiments are designed to answer the following research questions (RQs) systematically.

- **RQ1:** How does GLINT compare against existing LLM-based and PLM-based DocRE models?
- **RQ2:** What is the individual contribution of each component in GLINT?
- **RQ3:** Does the LLM backbone internalize rule-based reasoning through end-to-end training?
- **RQ4:** Does the negation gate learn meaningful and interpretable negation patterns?

| Model                        | DocRED       |              |              |              | Re-DocRED    |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                              | Dev          |              | Test         |              | Dev          |              | Test         |              |
|                              | F1           | Ign F1       | F1           | Ign F1       | F1           | Ign F1       | F1           | Ign F1       |
| PromptRE (Gao et al., 2024)  | -            | -            | -            | -            | 10.55        | 9.03         | -            | -            |
| ChatGPT (Han et al., 2023)   | 32.21        | -            | -            | -            | 28.89        | -            | -            | -            |
| DocGNRE (Li et al., 2023)    | 13.84        | 13.65        | 13.93        | 13.67        | 11.18        | 11.10        | 11.12        | 11.04        |
| D-F (Xue et al., 2024)       | 46.38        | 44.77        | 47.08        | 45.30        | 54.22        | 53.48        | 53.33        | 52.50        |
| D-R-F (Xue et al., 2024)     | 45.77        | 44.32        | 47.50        | 45.98        | 56.58        | 56.10        | 54.84        | 54.35        |
| AutoRE (Xue et al., 2024)    | 47.17        | 45.58        | 47.15        | 45.45        | 60.17        | 59.25        | 59.29        | 58.33        |
| EP-RSR (Zhang et al., 2025a) | 53.77        | 51.25        | 54.57        | 51.77        | 65.14        | 63.93        | 64.24        | 63.03        |
| LMRC (Li et al., 2026)*      | 61.01        | <u>59.13</u> | 60.84        | 58.77        | -            | -            | 72.80        | 72.21        |
| GLINT (LLaMA3-8B-Instruct)   | <b>61.75</b> | <b>59.42</b> | <b>62.38</b> | <u>59.86</u> | <b>77.47</b> | <b>76.32</b> | <b>77.33</b> | <b>76.19</b> |
| GLINT (LLaMA3.1-8B-Instruct) | <u>61.30</u> | 59.03        | <u>62.29</u> | <b>59.88</b> | <u>76.63</u> | <u>75.34</u> | <u>76.28</u> | <u>74.97</u> |

Table 1: Performance on DocRED and Re-DocRED. Baseline results are taken from Zhang et al. (2025a), except the result marked with \*, which is from their original paper. Best results are in **bold** and second-best are underlined.

| Model                        | DWIE         |              |              |              |
|------------------------------|--------------|--------------|--------------|--------------|
|                              | Dev          |              | Test         |              |
|                              | F1           | Ign F1       | F1           | Ign F1       |
| ChatGPT (Han et al., 2023)   | -            | -            | 26.72        | -            |
| DocGNRE (Li et al., 2023)    | 11.85        | 10.55        | 13.12        | 10.73        |
| AutoRE (Xue et al., 2024)    | 56.53        | 52.74        | 56.38        | 49.31        |
| EP-RSR (Zhang et al., 2025a) | 70.23        | 66.32        | 74.38        | 69.56        |
| GLINT (LLaMA3-8B-Instruct)   | <b>74.97</b> | <b>70.68</b> | <u>78.13</u> | <u>72.65</u> |
| GLINT (LLaMA3.1-8B-Instruct) | <u>74.05</u> | <u>68.79</u> | <b>80.69</b> | <b>75.50</b> |

Table 2: Performance on the DWIE dataset. Results of the other models are referred from Zhang et al. (2025a). Best results are in **bold** and second-best are underlined.

## 5.1 Experiment Setup

**Datasets.** We evaluate our approach on three DocRE datasets: DocRED (Yao et al., 2019), its re-annotated counterpart Re-DocRED (Tan et al., 2022) that addresses the false-negative issue in DocRED, and DWIE (Zaporojets et al., 2021), a manually annotated document-level dataset. The detailed statistics of the datasets can be found in Appendix D.

**Evaluation Metrics.** Following Yao et al. (2019), we report F1 and Ign F1 as our primary evaluation metrics, where Ign F1 measures the F1 score after removing relational facts that overlap between the training and evaluation sets.

**Implementation Details.** We use Llama3-8B-Instruct (Grattafiori et al., 2024) as our backbone LLM, chosen for fair comparison with recent LLM-based DocRE methods. The model is fine-tuned with QLoRA (Detrmers et al., 2023). All experiments are conducted on a single NVIDIA GPU (RTX 6000 Ada, 48GB; or RTX PRO 6000 Blackwell, 96GB). Full hyperparameter settings and prompt templates are provided in Appendix E and Appendix K.

**Baselines.** We compare our method against two categories of LLM-based DocRE approaches: those operating without task-specific fine-tuning and those fine-tuned on DocRE supervision. For the non-fine-tuned category, we include PromptRE (Gao et al., 2024), DocGNRE (Li et al., 2023), and the few-shot ChatGPT (Han et al., 2023) baseline. For the fine-tuned category, we adopt LMRC (Li et al., 2026), AutoRE (Xue et al., 2024) with its Document-Facts (D-F) and Document-Relation-Facts (D-R-F) paradigms, and EP-RSR (Zhang et al., 2025a) with its Entity Pair-Relation-Fact (EPRF) paradigm. We further include PLM-based baselines: backbones GAIN (Zeng et al., 2020) and ATLOP (Zhou et al., 2021), and rule-enhanced extensions LogiRE (Ru et al., 2021), MILR (Fan et al., 2022), and JMRL (Qi et al., 2024).

## 5.2 Main Results (RQ1)

Tables 1 and 2 report results on DocRED, Re-DocRED and DWIE datasets, with all baselines using LLaMA3-8B-Instruct. With LLaMA3-8B-Instruct, GLINT consistently outperforms prior LLM-based methods, surpassing the strongest baseline by +1.54 F1 on DocRED test, +4.53 on Re-DocRED test, and +3.75 on DWIE test. These consistent gains support the effectiveness of end-to-end logical-rule integration over pipeline-based LLM approaches. We further evaluate GLINT using LLaMA3.1-8B-Instruct and obtain comparable results overall. LLaMA3-8B-Instruct performs slightly better on DocRED and Re-DocRED, whereas LLaMA3.1-8B-Instruct achieves the best DWIE test result, increasing the margin over the strongest baseline to +6.31 F1. This consistency

| Method                    | Test (using test <sup>†</sup> ) |              |
|---------------------------|---------------------------------|--------------|
|                           | F1                              | Ign F1       |
| GAIN                      | 41.68                           | 41.26        |
| LogiRE-GAIN               | 41.89                           | 41.53        |
| MILR-GAIN                 | 43.17                           | 42.89        |
| JMRL-GAIN                 | <u>49.58</u>                    | 47.85        |
| ATLOP                     | 41.95                           | 41.67        |
| LogiRE-ATLOP              | 42.73                           | 42.47        |
| MILR-ATLOP                | 44.72                           | 44.30        |
| JMRL-ATLOP                | 47.54                           | 47.32        |
| GLINT <i>w/o negation</i> | 48.81                           | 48.49        |
| <b>GLINT</b>              | <b>49.78</b>                    | <b>49.58</b> |

Table 3: Comparison with PLM-based DocRE: training on DocRED and testing on Re-DocRED. Baseline results are taken from Qi et al. (2024), and all baselines employ BERT<sub>base</sub> for encoding. Both GLINT variants use LLaMA3.1-8B-Instruct. Best results are in bold and second-best are underlined.

suggests that the gains are driven primarily by our framework rather than by the choice of underlying model. We additionally verify generalization across backbone families and scales in Appendix J.

**Comparison with PLM-based DocRE.** As an additional comparison, we further investigate whether GLINT matches PLM-based neuro-symbolic approaches. Table 3 contrasts GLINT with representative rule-based DocRE frameworks built on PLM backbones. Under the DocRED setting identical to that of JMRL for a fair comparison, the *w/o negation* variant, which removes only the per-atom negation gates while keeping all other GLINT components fixed, achieves 48.81 F1, outperforming JMRL-ATLOP’s 47.54 but remaining below JMRL-GAIN’s 49.58. Adding the negation gate improves F1 by 0.97 to 49.78. While prior LLM-based DocRE models have largely failed to match the performance of PLM-based models, GLINT closes this gap, outperforming JMRL-GAIN, the strongest baseline among the PLM-based methods, by 0.20 F1.

### 5.3 Ablation Study (RQ2)

We ablate four components of GLINT to isolate their contributions, with results reported in Table 4. The dominant contributor is the  $\phi$  formulation (Eq. 13). Removing the non-candidate branch  $(1 - \hat{m}_{h,t}) \tilde{z}^{\text{DCF}}$  drops F1 by 4.45, confirming that this branch is essential for recovering relations on pairs that pipelines would discard. Removing the rule reasoner module drops F1 by 3.61, showing that chain-based composition adds expressive power beyond the LLM backbone alone. Within the

| Model             | F1           | Ign F1       | $\Delta$ F1 |
|-------------------|--------------|--------------|-------------|
| <b>GLINT</b>      | <b>77.47</b> | <b>76.32</b> | —           |
| w/o rule reasoner | 73.86        | 72.54        | −3.61       |
| w/o negation      | 76.88        | 75.61        | −0.59       |
| w/o st-gumbel     | 76.16        | 75.93        | −1.31       |
| w/o $\phi$        | 73.02        | 72.05        | −4.45       |

Table 4: Ablation study on the Re-DocRED dev set.  $\Delta$  F1 reports the F1 drop relative to the full GLINT.

| Setting                 | Dev          |              | Test         |              |             |
|-------------------------|--------------|--------------|--------------|--------------|-------------|
|                         | F1           | Ign F1       | F1           | Ign F1       | $\Delta$ F1 |
| <b>GLINT</b>            | <b>77.47</b> | <b>76.32</b> | <b>77.33</b> | <b>76.19</b> | +5.08       |
| GLINT <sub>zero</sub>   | 74.32        | 73.96        | 75.85        | 75.62        | +3.60       |
| GLINT <sub>norule</sub> | 73.86        | 72.54        | 72.25        | 71.93        | —           |

Table 5: Internalization analysis on Re-DocRED. The rightmost column reports F1 gain over GLINT<sub>norule</sub>.

rule reasoner, the per-atom negation gate yields a smaller but distinct gain, with its removal reducing F1 by 0.59 points. The full model improves precision over the positive-only *w/o negation* variant from 78.65 to 80.66 (+2.01 points), while recall decreases from 75.19 to 74.51 (−0.68 points). This gain in precision, together with the modest recall cost, is consistent with negation suppressing unsupported relation predictions, as illustrated in Appendix H. Finally, removing the Straight-Through estimator on  $\hat{m}_{h,t}$  (Eq. 6), which routes gradient of  $\mathcal{L}_{\text{main}}$  back to the DCF, drops F1 by 1.31, showing that this gradient pathway helps the model handle pairs whose candidate status is ambiguous.

### 5.4 Rule Internalization Analysis (RQ3)

To examine whether end-to-end training leads the LLM backbone to internalize rule-based reasoning, we compare three settings on Re-DocRED: (A) **GLINT**, our complete model with the rule module active, (B) **GLINT<sub>zero</sub>**, same model with rule scores set to zero at inference, and (C) **GLINT<sub>norule</sub>**, an identical architecture trained without the rule module. The key comparison is between (B) and (C). Both operate without the rule residual at inference, but only (B) is trained with the rule-augmented objective. During training, this objective updates the shared backbone through the direct GRE branch and the straight-through candidate-selection path, whereas (C) is trained without rule-based supervision. As shown in Table 5, (B) exceeds (C) by +3.60 F1 on the test set. Since both settings disable the rule residual at inference, this gain cannot be attributed to additional

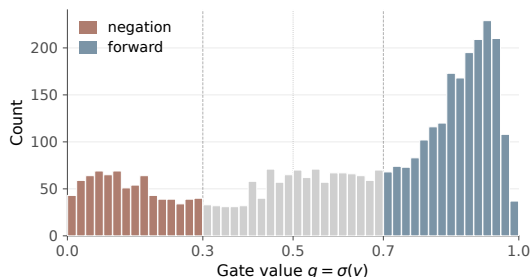


Figure 4: Distribution of learned negation gate values on Re-DocRED. Gates with red ( $g < 0.3$ ) implement  $\neg$  atom, while gates with blue ( $g > 0.7$ ) reproduce the standard positive atom.

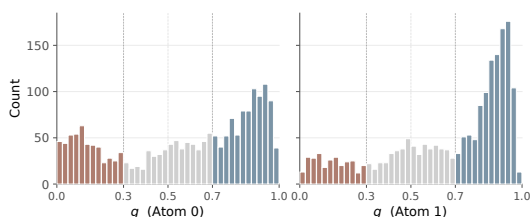


Figure 5: Per-atom gate distribution on Re-DocRED.

inference-time rule capacity. It instead indicates that the backbone retains patterns induced by joint rule-augmented training. Notably, 3.60 of the full model’s 5.08-point improvement over (C) remains after the rule scores are removed. The remaining 1.48-point gap between (A) and (B) suggests that explicit inference-time rule reasoning provides a substantial, complementary benefit. Appendix G further shows that post-hoc training of the rule module on a frozen backbone does not recover the end-to-end model, reinforcing the importance of joint adaptation.

### 5.5 Negation Gate Analysis (RQ4)

We investigate whether the gate actually learns negation patterns. A separate gate is learned at every atomic position in the rule module Figure 4 shows two distinct modes in the learned gate values. Specifically, 51.2% of the gates settle in the forward regime ( $g > 0.7$ ), while 20.1% settle in the negation regime ( $g < 0.3$ ). Together, 71.3% of the gates lie in one of these two polarity-specific regimes, showing that most gates move away from the neutral initialization ( $g = 0.5$ ) rather than remaining ambiguous. Figure 5 further shows an asymmetry across chain positions. Atom 0 enters the negation regime in 25.8% of cases, compared with 14.4% for atom 1. This 11.4-point gap indicates a position-specific preference for placing negated conditions at the earlier hop, rather than a uniform tendency toward negation across the chain.

The negation gates and signed rule confidences play distinct roles. The gates determine whether individual body literals are positive or negated, whereas the signed confidences determine whether a satisfied rule increases or decreases the head logit. Thus, the gate distributions characterize learned literal polarity rather than the sign of the final rule contribution. The mixed-polarity case in Appendix H illustrates their interaction, where a gate deep in the negation regime ( $g = 0.10$ ) combines with a negative rule weight to suppress an unsupported relation prediction.

## 6 Conclusion

We propose GLINT, an end-to-end DocRE framework that internalizes negation-gated logical rules into an LLM via the Fused Fact tensor  $F^+$ , recovering missing facts through logical reasoning while suppressing spurious predictions. Experiments on three benchmarks demonstrate that GLINT consistently outperforms strong baselines, and our analyses confirm that the LLM backbone internalizes rule-induced reasoning patterns while the learned negation gates capture logical structures beyond positive-only formulations. Future work will extend GLINT to broader neuro-symbolic settings with deeper coupling between symbolic rules and LLM representations, consistent with recent work (Lee et al., 2026b) that leverages model-internal probabilistic signals for fine-grained reasoning supervision.

### Limitations

We discuss two limitations of GLINT. On one hand, we restrict our experiments to 8B-parameter backbones (LLaMA3-8B-Instruct and LLaMA3.1-8B-Instruct) to ensure fair comparison with LLM-based DocRE baselines. Whether the benefits of end-to-end rule internalization persist at larger scales remains an open question, particularly as larger LLMs may exhibit different over-prediction patterns and priors that the rule reasoner must adapt to. Conversely, whether the framework can be distilled into smaller backbones for efficient deployment is worth exploring, since the rule reasoner provides an explicit symbolic structure that may transfer across model scales. On the other hand, as with other LLM-based DocRE methods, GLINT requires fine-tuning an LLM backbone, which incurs non-trivial computational cost at training time. While this cost remains within the range of LLM

fine-tuning under QLoRA, it may limit applicability in settings with strict budgets.

## Ethical considerations

This work complies with the ACL Code of Ethics<sup>2</sup>. We conduct all experiments on three publicly available DocRE benchmarks — DocRED, ReDocRED, and DWIE — which are widely used in the community and contain no private or personally identifiable information beyond what is already publicly available in their source corpora. We use LLaMA-3 and LLaMA-3.1 under the Meta Llama 3 Community License<sup>3</sup>. While GLINT improves the logical consistency of relation predictions, document-level relation extraction systems may still produce factually incorrect outputs; downstream deployment in high-stakes settings should therefore incorporate human verification. We will release our code and trained checkpoints upon acceptance to support reproducibility.

## Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) in 2026 (Grant No. RS-2024-00457435). This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2026-RS-2026-25615817, AI Star Fellowship Support Program). This work was supported by the National Natural Science Foundation of China (No. 62306287).

## References

- Serge Abiteboul, Richard Hull, and Victor Vianu. 1995. *Foundations of databases*. Addison-Wesley Reading.
- Hye-Yoon Baek, Jinho Choi, Jimyeung Seo, Xiongnan Jin, Dongcheon Lee, and Byungkook Oh. 2025. Relation-faceted graph pooling with LLM guidance for dynamic span-aware information extraction. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 109–118.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.
- Shengda Fan, Shasha Mo, and Jianwei Niu. 2022. Boosting document-level relation extraction by mining and injecting logical rules. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10311–10323.
- Chufan Gao, Xulin Fan, Jimeng Sun, and Xuan Wang. 2024. Promptre: Weakly-supervised document-level relation extraction via prompting-based data programming. In *Proceedings of the 1st workshop on towards knowledgeable language models (knowLLM 2024)*, pages 132–145.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*, 48.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.
- Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.
- Geunyeong Jeong, Juoh Sun, Seonghee Lee, and Harksoo Kim. 2025. STEAM: A semantic-level knowledge editing framework for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 11008–11023.
- Sangjun Ji, Sangjune Kim, Youngho Lee, Bonyou Koo, Xiongnan Jin, and Byungkook Oh. 2026. Joint global-local representations via relation-entity pair encoding for hyper-relational knowledge graphs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 2050–2061.
- Dongcheon Lee, Ji-Yeon Park, Hye-Yoon Baek, Jimyeung Seo, Seyeong Kim, and Byungkook Oh. 2026a. Diversifying differentiable graph retrieval with topic-adaptive multi-intent learning. In *Proceedings of the ACM Web Conference (WWW)*, pages 2273–2284.
- Nakyung Lee, Sangwoo Hong, and Jungwoo Lee. 2026b. Efficient process reward modeling via contrastive mutual information. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 37603–37617.
- Junpeng Li, Zixia Jia, and Zilong Zheng. 2023. Semi-automatic data enhancement for document-level relation extraction with distant supervision from large

<sup>2</sup><https://www.aclweb.org/portal/content/acl-code-ethics>

<sup>3</sup><https://llama.meta.com/llama3/license>

- language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5495–5505.
- Xingzuo Li, Kehai Chen, Yunfei Long, and Min Zhang. 2026. Llm with relation classifier for document-level relation extraction. *IEEE Transactions on Big Data*.
- Yichun Liu, Zizhong Zhu, Xiaowang Zhang, Zhiyong Feng, Daoqi Chen, and Yaxin Li. 2023. Document-level relationship extraction by bidirectional constraints of beta rules. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2266.
- Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.
- Youmi Ma, An Wang, and Naoaki Okazaki. 2023a. Dream: Guiding attention with evidence for improving document-level relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1971–1983.
- Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023b. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the association for computational linguistics: EMNLP 2023*, pages 10572–10601.
- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR*.
- Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1546–1557.
- Kunxun Qi, Jianfeng Du, and Hai Wan. 2024. End-to-end learning of logical rules for enhancing document-level relation extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7247–7263.
- Chris Quirk and Hoifung Poon. 2017. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1171–1182.
- Dongyu Ru, Changzhi Sun, Jiangtao Feng, Lin Qiu, Hao Zhou, Weinan Zhang, Yong Yu, and Lei Li. 2021. Learning logic rules for document-level relation extraction. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pages 1239–1250.
- Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred-addressing the false negative problem in relation extraction. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 8472–8487.
- Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 3534–3547.
- Lilong Xue, Dan Zhang, Yuxiao Dong, and Jie Tang. 2024. Autore: Document-level relation extraction with large language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: System demonstrations)*, pages 211–220.
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *Proceedings of the Third International Conference on Learning Representations (ICLR)*.
- Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 764–777.
- Klim Zaporozhets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. Dwie: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 58(4):102563.
- Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 1630–1640.
- Fu Zhang, Hongsen Yu, Jingwei Cheng, and Huangming Xu. 2025a. Entity pair-guided relation summarization and retrieval in llms for document-level relation extraction. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4022–4037.
- Kunli Zhang, Pengcheng Wu, Bohan Yu, Kejun Wu, Aoze Zheng, Xiyang Huang, Chenkang Zhu, Min Peng, Hongying Zan, and Yu Song. 2025b. Cadrl: Document-level relation extraction via context-aware differentiable rule learning. In *Proceedings of the 31st International Conference on Computational Linguistics, (COLING)*, pages 8272–8284.
- Yudi Zhang, Pei Xiao, Lu Wang, Chaoyun Zhang, Meng Fang, Yali Du, Yevgeniy Puzyrev, Randolph Yao, Si Qin, Qingwei Lin, and 1 others. 2025c. Ruag:

Learned-rule-augmented generation for large language models. In *International Conference on Learning Representations*, volume 2025, pages 37697–37720.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14612–14620.

## A Preliminary Experiments

We provide the details of our preliminary experiments supporting the problem analysis in Section 1.

### LLM-based DocRE Formulations on DocRED.

We examine three formulations that vary in input granularity and supervision, and report P, R, and F1 for each (Figure 2).

In **(a) document-level zero-shot**, the LLM extracts all entity pairs and their relations in a single response. Both precision and recall remain low, indicating that this joint extraction exceeds the LLM’s single-pass capacity. This is consistent with recent findings that LLMs struggle to distinguish no-relation cases from predefined labels under prompting (Ma et al., 2023b; Wan et al., 2023). Setting **(b) pair-level zero-shot** keeps the zero-shot regime but feeds the LLM the document along with a candidate entity pair to predict the relations between them. Recall more than doubles over (a), but precision drops sharply, reflecting over-prediction when forced to commit to a per-pair output.

Finally, **(c) pair-level fine-tuned binary classification** fine-tunes the LLM to predict whether a given pair holds any relation (has\_relation) or not (no\_relation). Positive and negative pairs are sampled at a 1:3 ratio per document, since over 97% of pairs in DocRED are labeled NA. This setting mirrors our DCF module and yields high recall (89.99) but low precision (30.96), motivating a separate downstream relation extractor.

**Pipeline Bottleneck Analysis.** To empirically ground the design of GLINT, we conduct a pre-experiment that isolates where errors accumulate in a naive two-stage pipeline. We train two modules independently: the **candidate filter** extracts the set of entity pairs that bear at least one relation, and the **relation extractor** predicts the relations of given entity pairs, taking only positive (gold) pairs as input. A **cascaded** setting then feeds the filter’s extracted pairs directly into the extractor at inference.

| Setting            | dev   |       |       |
|--------------------|-------|-------|-------|
|                    | P     | R     | F1    |
| Candidate filter   | 44.82 | 66.56 | 53.57 |
| Relation extractor | 89.93 | 89.52 | 89.72 |
| Cascaded           | 41.27 | 61.68 | 49.45 |

Table 6: Pipeline bottleneck analysis on DocRED. All modules are Llama-3.1-8B-Instruct fine-tuned with QLoRA (Detrmers et al., 2023). The candidate filter is evaluated on entity-pair set extraction; the others on relation extraction.

Table 6 reports results on DocRED dev set. The LLM backbone itself is not the limiting factor: given gold positive pairs, the relation extractor performs strongly, indicating that the LLM classifies relations competently once the candidate space is correctly constrained. The collapse occurs entirely at the selection boundary. Since any pair the candidate filter misses cannot be re-introduced downstream, its recall structurally upper-bounds the pipeline. The cascaded setting therefore degrades sharply relative to the relation extractor with gold pairs, and the drop is not confined to recall: precision also collapses, showing that the filter’s false positives cascade into spurious relation predictions and that selection errors compound rather than cancel downstream.

---

### Algorithm 1: GRE forward pass with sequence packing

---

**Input:** instruction  $I$ , document  $d$ , entity set  $\mathcal{E}_d$ , relation vocabulary  $\mathcal{R}$ , candidate set  $\mathcal{C}$ , GRE mini-batch size  $M$ , relation token IDs  $\mathcal{I}_{\mathcal{R}}$

**Output:** relation logits  $\{\mathbf{z}_{h,t}^{\text{GRE}}\}_{(h,t) \in \mathcal{C}}$

```

1 Tokenize shared prefix  $p \leftarrow (I, d, \mathcal{E}_d, \mathcal{R})$ 
2 Partition  $\mathcal{C}$  into mini-batches  $\mathcal{B}_1, \dots, \mathcal{B}_K$  of size at most  $M$ 
3 for  $k = 1$  to  $K$  do
4   Build packed input  $s_k \leftarrow [p; q_1; \dots; q_{|\mathcal{B}_k|}]$ ,
5   where  $q_i = \text{Query}(h_i, t_i) \parallel [\text{REL}]_i$  for  $(h_i, t_i) \in \mathcal{B}_k$ 
6   Build block-causal attention  $A_k$ :
7   - prefix  $p$ : lower-triangular (causal)
8   - each query block  $i$ : full attention to  $p$ , lower-triangular within itself,
9     zero attention to other query blocks
10   $H_k \leftarrow \text{LLM}(s_k, A_k)$  // single forward pass
11  for each pair  $(h_i, t_i) \in \mathcal{B}_k$  do
12     $\mathbf{h}_i^{[\text{REL}]} \leftarrow H_k[\text{pos}([\text{REL}]_i)]$ 
13     $\mathbf{z}_{h_i, t_i}^{\text{GRE}} \leftarrow \text{LMHead}(\mathbf{h}_i^{[\text{REL}]})[\mathcal{I}_{\mathcal{R}}]$ 
14  end
15 end
16 return  $\{\mathbf{z}_{h,t}^{\text{GRE}}\}_{(h,t) \in \mathcal{C}}$ 

```

---

Motivated by this, GLINT prioritizes recall in the DCF module (Section 4.1) to avoid imposing an upstream recall ceiling on the overall system.

## B Sequence Packing in GRE Module

Algorithm 1 details how the GRE module processes candidate pairs  $\mathcal{C}$  via sequence packing and block-diagonal attention. Within each packed forward pass, the prefix tokens are processed once, rather than recomputed per pair as in a naïve per-pair forward, while the block-diagonal mask ensures that each pair’s logits depend only on the prefix and its own query. Position IDs are assigned contiguously across the packed sequence; independence between queries is enforced by the mask rather than by positional separation.

## C Formalization of Rule Reasoner

This appendix provides the full composition and aggregation formulas of the rule reasoner in Section 4.3, deferred from the main text. We follow the notation of Section 4.3.

**Pre-gate atom from  $\mathbf{F}^+$ .** Each hop of a chain rule must select one of the  $2n + 1$  predicates in  $\mathcal{R}$ . To make this choice differentiable, we replace hard predicate selection with a learnable distribution over  $\mathcal{R}^+$ . For each target relation  $r_k$ , rule  $j$ , and hop  $\ell$ , we maintain a learnable score vector  $\mathbf{u}_{k,j}^{(\ell)} \in \mathbb{R}^{2n+1}$ , where the  $p$ -th entry expresses the raw preference for selecting predicate  $p \in \mathcal{R}^+$  at this position. The pre-gate atom at pair  $(x, y)$  is the expected  $\mathbf{F}^+$  score under the softmax of  $\mathbf{u}_{k,j}^{(\ell)}$ :

$$\tilde{a}_{k,j}^{(\ell)}(x, y) = \sum_{p \in \mathcal{R}^+} [\text{softmax}(\mathbf{u}_{k,j}^{(\ell)})]_p \cdot \mathbf{F}^+(x, y, p), \quad (15)$$

which lies in  $[0, 1]$  since the softmax output is a probability distribution and  $\mathbf{F}^+(x, y, \cdot) \in [0, 1]^{2n+1}$ .

**Negation-gated literal.** The pre-gate atom  $\tilde{a}_{k,j}^{(\ell)}$  captures only positive predicate semantics. To enable negated literals, we attach a per-atom negation gate  $g_{k,j}^{(\ell)} = \sigma(v_{k,j}^{(\ell)}) \in (0, 1)$  on a learnable scalar  $v_{k,j}^{(\ell)} \in \mathbb{R}$ , and form the gated literal as (restating Eq. 9)

$$a_{k,j}^{(\ell)}(x, y) = g_{k,j}^{(\ell)} \tilde{a}_{k,j}^{(\ell)}(x, y) + (1 - g_{k,j}^{(\ell)}) (1 - \tilde{a}_{k,j}^{(\ell)}(x, y)). \quad (16)$$

Setting  $g \rightarrow 1$  recovers the standard positive atom  $a \rightarrow \tilde{a}$ , while  $g \rightarrow 0$  yields  $a \rightarrow 1 - \tilde{a}$ , which is the soft-logic encoding of  $\neg \tilde{a}$ . Since both  $\tilde{a}_{k,j}^{(\ell)}(x, y)$  and  $g_{k,j}^{(\ell)}$  lie in  $[0, 1]$ , the gated literal remains in  $[0, 1]$ , allowing it to act as a soft truth value.

### Chain Composition over Intermediate Entities.

A depth- $L$  chain rule for relation  $r_k$ , rule  $j$ , composes  $L$  gated literals along an entity path  $x_0 \rightarrow x_1 \rightarrow \dots \rightarrow x_L$  as a soft conjunction, with intermediate entities  $x_1, \dots, x_{L-1}$  marginalized over  $\mathcal{E}_d$ . Collecting the literal values at all entity pairs into a matrix  $\mathbf{A}_{k,j}^{(\ell)} \in [0, 1]^{n_d \times n_d}$  with entries

$$[\mathbf{A}_{k,j}^{(\ell)}]_{x,y} = a_{k,j}^{(\ell)}(x, y), \quad (17)$$

chain composition reduces to a matrix product:

$$\mathbf{C}_{k,j} = \mathbf{A}_{k,j}^{(1)} \mathbf{A}_{k,j}^{(2)} \dots \mathbf{A}_{k,j}^{(L)} \in \mathbb{R}^{n_d \times n_d}, \quad (18)$$

whose  $(h, t)$ -entry

$$[\mathbf{C}_{k,j}]_{h,t} = \sum_{x_1, \dots, x_{L-1} \in \mathcal{E}_d} \prod_{\ell=1}^L a_{k,j}^{(\ell)}(x_{\ell-1}, x_\ell), \quad (19)$$

with  $x_0 = h$  and  $x_L = t$ . Each length- $L$  path from  $h$  to  $t$  contributes the product of its  $L$  gated literals, and the chain score  $[\mathbf{C}_{k,j}]_{h,t}$  sums these contributions over all choices of intermediate entities.

**Aggregation across  $N$  rules.** For each target relation  $r_k$ , the  $N$  rule-specific chain scores  $\{[\mathbf{C}_{k,j}]_{h,t} * h, t\}_{j=1}^N$  are combined into a single rule logit via signed confidences. We attach a learnable scalar  $\beta_{k,j} \in \mathbb{R}$  to each rule and define its signed confidence as

$$\alpha_{k,j} = \tanh(\beta_{k,j}) \in (-1, 1), \quad (20)$$

where the sign and magnitude of  $\alpha_{k,j}$  encode the direction and strength of rule  $j$ ’s contribution to predicting  $r_k$ . The rule logit for pair  $(h, t)$  and relation  $r_k$  is then the signed sum of chain scores:

$$z_{h,t,k}^{\text{rule}} = \sum_{j=1}^N \alpha_{k,j} \cdot [\mathbf{C}_{k,j}]_{h,t}. \quad (21)$$

which enters the final rule-augmented prediction  $\phi_{h,t,k}$  as a residual term (Eq. 13).

## D Details and Statistics of Datasets

We evaluate our proposed GLINT on three widely used benchmark datasets in DocRE, which are shown as follows:

| Dataset   | Split | #Doc. | #Rel. | #Ent.  | #Facts. |
|-----------|-------|-------|-------|--------|---------|
| DocRED    | train | 3,053 |       | 59,493 | 38,180  |
|           | dev   | 998   | 96    | 19,578 | 12,323  |
|           | test  | 1,000 |       | 19,539 | -       |
| Re-DocRED | train | 3,053 |       | 59,359 | 85,932  |
|           | dev   | 500   | 96    | 9,684  | 17,284  |
|           | test  | 500   |       | 9,779  | 17,448  |
| DWIE      | train | 602   |       | 16,494 | 14,403  |
|           | dev   | 98    | 65    | 2,785  | 2,624   |
|           | test  | 99    |       | 2,623  | 2,495   |

Table 7: Statistics on the datasets, where Doc. is short for documents, Rel. for relation types, Ent. for entities, and Fact for relational facts.

- **DocRED** (Yao et al., 2019) is a common benchmark for DocRE, comprising a diverse set of documents constructed from Wikipedia. Since relation evidence is often distributed across multiple sentences, DocRED requires models to capture document-level context, coreference information, and inter-sentence dependencies.
- **Re-DocRED** (Tan et al., 2022) is a re-annotated version of the original DocRED dataset. It was introduced to address the false-negative issue in DocRED, where some valid relation facts are missing from the original annotations. By providing more complete relation annotations, Re-DocRED offers a more reliable evaluation setting for document-level relation extraction.
- **DWIE** (Zaporojets et al., 2021) is a manually annotated document-level information extraction dataset. It contains annotations for entities, coreference, and relations within documents. These rich document-level annotations make DWIE a useful benchmark for evaluating relation extraction in a broader information extraction setting.

The detailed statistics of the datasets can be found in Table 7.

## E Hyperparameter Settings

For reproducibility, Table 8 reports the detailed hyperparameter settings used in our experiments. We use the same hyperparameters for both LLaMA3-8B-Instruct and LLaMA3.1-8B-Instruct backbones. We adopt LoRA (Hu et al., 2022) for parameter-efficient fine-tuning and AdamW (Loshchilov and

| Parameter                                       | Value                 |
|-------------------------------------------------|-----------------------|
| Number of rules ( $N$ )                         | 20                    |
| Chain length ( $L$ )                            | 2                     |
| Candidate top- $k$                              | 50                    |
| Projection dim                                  | 256                   |
| LoRA ( $r, \alpha$ ), dropout                   | (16, 32), 0.05        |
| Quantization                                    | 4-bit NF4, bf16       |
| Optimizer                                       | AdamW                 |
| Weight decay                                    | 0.01                  |
| Learning rate (LoRA)                            | 2e-4                  |
| Learning rate (DCF head)                        | 1e-3                  |
| Learning rate (rule module)                     | 3e-1                  |
| Learning rate ( $F^+$ scaler)                   | 1e-2                  |
| Warmup ratio                                    | 0.06                  |
| Max gradient norm                               | 1.0                   |
| Epochs                                          | 3                     |
| Batch size                                      | 1                     |
| Gumbel temperature                              | 1.0 $\rightarrow$ 0.1 |
| Positive weight (DCF, GRE)                      | (8.0, 10.0)           |
| Loss weights ( $\lambda_{RCP}, \lambda_{aux}$ ) | (0.1, 1.0)            |

Table 8: Hyperparameter settings.

Hutter, 2017) as the optimizer. Hyperparameters are tuned on the development set to maximize F1.

## F Empirical Analysis of DCF Module

### F.1 Architectural Design Choices

The DCF module assigns a relation-likelihood score to every entity pair in a document. Knowledge graph completion (KGC) methods provide a natural starting point for scoring entity pairs, but they differ substantially in computational cost and the assumptions they impose on relation structure. We adopt DistMult (Yang et al., 2015) for the DCF module because it offers the lowest per-pair scoring cost among standard KGC methods while remaining expressive enough for filtering. However, the standard DistMult formulation is symmetric in head and tail, which is incompatible with directional DocRE relations. We therefore introduce two architectural modifications.

We empirically justify two design choices of our DCF module on Re-DocRED: (i) using separate projections for head and tail entities rather than the classical symmetric DistMult formulation, and (ii) initializing relation embeddings from mean-pooled LLM token embeddings rather than random vectors. We report pair selection precision, recall, F1 and the raw false positive (FP) counts on the development set after three training epochs with the LLaMA3.1-8B-Instruct, as summarized in Table 9. **Asymmetric over symmetric scoring.** Classical DistMult scores entity pairs with a single shared bilinear form, which is inherently symmetric in

| Projection | Relation Init           | P            | R            | F1           | FP           |
|------------|-------------------------|--------------|--------------|--------------|--------------|
| Symmetric  | Random                  | 42.78        | 89.81        | 57.95        | 16,052       |
|            | LLM-token               | 43.70        | 90.41        | 58.92        | 15,561       |
| Asymmetric | Random                  | 58.16        | 90.10        | 70.69        | 8,662        |
|            | <b>LLM-token (Ours)</b> | <b>63.98</b> | <b>89.73</b> | <b>74.70</b> | <b>6,751</b> |

Table 9: Ablation on DCF module design choices on Re-DocRED. The bottom row is our final configuration.

| Method                   | P    | R     | F1   |
|--------------------------|------|-------|------|
| Classical DistMult       | 1.06 | 6.35  | 1.81 |
| <b>Ours (asymmetric)</b> | 1.48 | 29.18 | 2.82 |

Table 10: Relation classification performance of the DCF module on Re-DocRED dev set.

head and tail and therefore fundamentally incompatible with DocRE, where most relations are directional. We instead apply two separate learnable projections for head and tail mentions, breaking the symmetry by construction. As shown in Table 9, this change yields the dominant improvement in F1. Recall remains nearly unchanged across all four configurations, while FPs more than double under the symmetric formulation. The symmetric scoring does not drop true positive pairs, but rather amplifies spurious predictions, exacerbating the over-prediction tendency of LLM-based DocRE defined in Section 1 rather than alleviating it.

Furthermore, Table 10 shows that both formulations achieve very low performance on relation classification (*which relation*), with a marginal 0.01 F1 gap, while differing substantially on pair selection (*any relation*), as shown in Table 9. This confirms the role of the DCF module as a filter rather than a classifier: pair selection is handled here, while fine-grained relation classification is left to the downstream GRE module.

**LLM token over random initialization.** We initialize each relation embedding by mean-pooling the LLM’s token embeddings for the corresponding label tokens. Unlike random initialization, which is entirely dependent on the training corpus, this design transfers prior knowledge from the LLM, providing the DCF module with a semantically grounded, corpus-independent starting point. The effect is consistent but more modest than the projection choice, and is more pronounced when combined with asymmetric projections, where the model has the capacity to exploit this prior.

## F.2 Sensitivity to Positive Weight

The DCF module is trained with binary cross-entropy on highly imbalanced candidate pair dis-

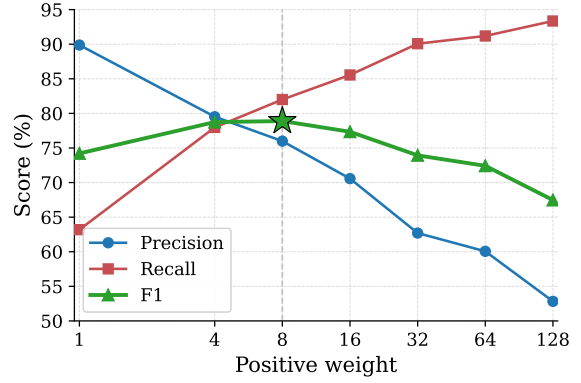


Figure 6: Precision, recall, and pair-level F1 on the DCF module on Re-DocRED across different positive weight values. The star marker indicates our adopted value.

tributions. To balance precision and recall in candidate selection, we sweep positive weight (pw) values from 1 to 128 on Re-DocRED. Figure 6 shows that the precision-recall trade-off is sharply controlled by this weight. Low values yield high precision but miss many candidates, while high values produce over-prediction. Pair-level F1 is nearly identical at  $pw = 4$  (78.74) and  $pw = 8$  (78.89). We adopt  $pw = 8$  as recall is more critical than precision in the candidate filtering stage.

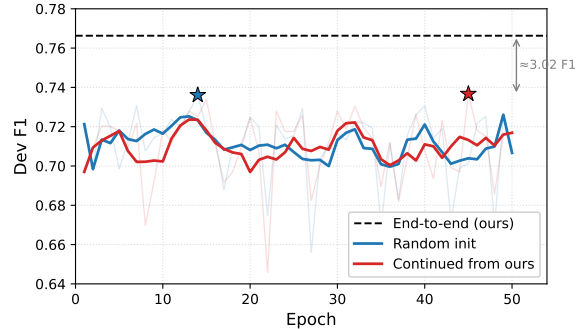


Figure 7: Per-epoch dev F1 under two initializations on Re-DocRED. Faint lines show raw F1. Bold lines show the 5-epoch moving average. Stars mark each setting’s best epoch. The dashed line marks our full model performance (F1=0.7663).

## G Rule Module Training on a Frozen Backbone

Our internalization analysis in Section 5.4 shows that the LLM backbone absorbs rule signal during end-to-end training. We here verify this from the opposite direction by training the rule module alone on a frozen backbone. We freeze the backbone and train only the rule module for 50 epochs on Re-DocRED, under two initializations: (i) **Random init**: rule module reinitialized from scratch, and (ii) **Continued from ours**: rule module loaded from our checkpoint at epoch 3. The rule input  $F^+$  is precomputed from the deployed end-to-end

| Documents                                                                                                                                                                                                                                                                                                                                                                                                         | Rule Template                                                                                                                                                                         | Predictions (GLINT)             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| <p><b>Ross Patterson Alger</b> (August 20, 1920 – January 16, 1992) was a politician in the <b>Canadian</b> province of Alberta, who served as mayor of Calgary from 1977 to 1980. Born in Prelate, Saskatchewan, he moved to Alberta with his family in 1930s. ... His brother was <b>Harry Alger</b>. Alger died of cancer in 1992 ...</p>                                                                      | <p>country_of_citizenship(x, y)<br/> <math>\leftarrow</math> sibling(x, z) [g=0.79]<br/> <math>\wedge</math> country_of_citizenship(z, y) [g=0.94]<br/> <b>weight = + 0.973</b></p>   | <p>(a) Positive 2-hop chain</p> |
| <p>Wilfried "Willi" Schneider (born 13 March 1963 in Mediaș, Transylvania) is a <b>German</b> skeleton racer who competed from 1992 to 2002. ... After retiring from competition Schneider became a coach, leading the <b>Canadian</b> skeleton team to three medals at the 2006 Winter Olympics in Turin (a gold for <b>Duff Gibson</b>, a silver for Jeff Pain and a bronze for Melissa Hollingsworth), ...</p> | <p>official_language(x, y)<br/> <math>\leftarrow</math> INV(citizenship)(x, z) [g=0.88]<br/> <math>\wedge</math> NOT languages_spoken(z, y) [g=0.10]<br/> <b>weight = - 0.987</b></p> | <p>(b) Negation suppression</p> |

Figure 8: Case study on the Re-DocRED dev set. Each row shows (left) a source document, (center) a rule template extracted from the learned negation gates and rule weights, and (right) the rule’s instantiation on document entities. Black-labeled relations are gold body atoms, while the head relation involved by the rule is colored gray (derived under a positive rule weight, top) or red (suppressed under a negative rule weight, bottom). Gates with  $g$  close to 1 implement the forward atom, while gates close to 0 implement its negation.

checkpoint and cached for efficiency.

Figure 7 shows that both settings fall short of the full model by approximately 3 F1 points, regardless of initialization, and converge to nearly identical scores. Training loss remains flat across all 50 epochs while F1 fluctuates by  $\pm 0.02$  per epoch, indicating that the optimizer has converged. The results support two observations. First, the rule module was not undertrained at epoch 3: extending training for 50 epochs, even from a well-initialized checkpoint, does not improve performance. Second, the rule module’s capability cannot be recovered by post-hoc training on a frozen backbone, regardless of initialization. End-to-end training is thus essential for both components: the LLM absorbs rule signal, and the rule module relies on a backbone trained alongside it.

## H Case Study

Figure 8 presents two qualitative examples of rules learned by GLINT on Re-DocRED, illustrating the two distinct roles the rule reasoner plays in our framework. We follow the rule extraction procedure of JMRL (Qi et al., 2024) and additionally read the learned negation gates  $g_{k,j}^{(\ell)}$  at the atom level, assigning polarity information to each atom of the extracted rule body.

**Positive 2-hop chain (Figure 8(a)).** This case shows the rule reasoner acting as a *fact recoverer*. Both atoms settle in the forward regime ( $g > 0.7$ ) and the rule weight is strongly positive, indicating

that GLINT has confidently learned the transitive pattern that siblings share citizenship. The rule contributes a large positive residual to the prediction logit, allowing the head relation to be derived even when it is not stated directly in the text. This demonstrates how chain composition over  $F^+$  recovers implicit relations that single-pass LLM extraction would miss.

**Negation suppression (Figure 8(b)).** This case shows the rule reasoner acting as a *spurious-prediction suppressor*, a role that positive-only formulations cannot fulfill. The second atom’s gate falls deep into the negation regime ( $g=0.10$ ), flipping the literal to its negation, while the rule weight is strongly negative. The body is satisfied precisely when the corresponding relation is *absent* from the document, and in that case the negative residual pulls the head logit down. This is the mechanism by which GLINT counteracts the LLM’s tendency to over-predict relations from surface co-occurrence: when two entities co-appear in a document without their relation being substantively grounded, the negation rule fires and suppresses the spurious head. This pattern is structurally inexpressible under positive-only rule formulations, empirically validating the design of per-atom negation gates.

## I Sensitivity to Top-K

The DCF module retains at most top- $k$  candidate pairs per document. To examine how this

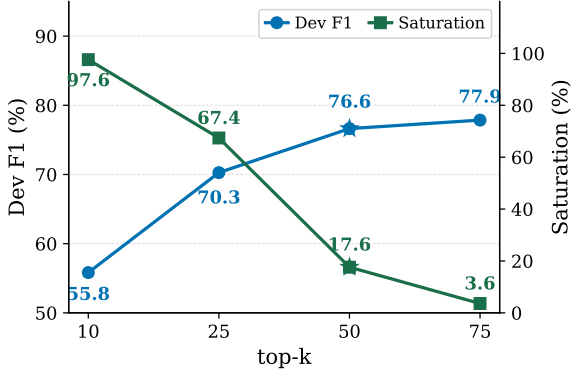


Figure 9: Dev F1 (left) and saturation rate (right) on Re-DocRED across top- $k$  values. The star marker indicates our adopted value ( $k = 50$ ).

hyperparameter affects performance, we sweep  $k \in \{10, 25, 50, 75\}$  on Re-DocRED. Figure 9 shows F1 alongside the saturation rate, fraction of documents reaching the top- $k$  limit. F1 increases substantially from  $k = 10$  (55.82) to  $k = 50$  (76.63), with only marginal further improvement at  $k = 75$  (77.86, +1.23 F1). Saturation drops from 97.6% at  $k = 10$  to 17.6% at  $k = 50$ , indicating that the selector itself determines the candidate set for most document at  $k = 50$  and beyond. The average candidate count increases only marginally beyond  $k = 50$  (9.91 at  $k = 10$ , 22.48 at  $k = 25$ , 31.80 at  $k = 50$ , 34.26 at  $k = 75$ ), further confirming that raising the cap yields diminishing utility. Training time at  $k = 75$  (18h) is 50% longer than at  $k = 50$  (12h), and exceeds 27h at  $k = 100$ , where we terminated the run. We adopt  $k = 50$  as the default, balancing F1 against computational cost.

## J Generalization Across Backbones

| Backbone                  | Dev          |              | Test         |              |
|---------------------------|--------------|--------------|--------------|--------------|
|                           | F1           | Ign F1       | F1           | Ign F1       |
| LMRC                      | –            | –            | 72.80        | 72.21        |
| <b>GLINT</b>              |              |              |              |              |
| <b>LLaMA3-8B-Instruct</b> | <b>77.47</b> | <b>76.32</b> | <b>77.33</b> | <b>76.19</b> |
| LLaMA3.1-8B-Instruct      | <u>76.63</u> | <u>75.34</u> | <u>76.28</u> | <u>74.97</u> |
| Qwen2.5-7B-Instruct       | 75.59        | 74.27        | 75.48        | 74.18        |
| Qwen2.5-3B-Instruct       | 73.45        | 72.08        | 73.70        | 72.45        |

Table 11: GLINT performance across backbone families and sizes on Re-DocRED, compared with LMRC, the strongest LLM-based pipeline baseline from Table 1. LMRC uses LLaMA3-8B-Instruct. Best results are in bold and second-best are underlined.

To examine whether GLINT depends on a particular backbone family or model scale, we additionally evaluate it with Qwen2.5-7B-Instruct and Qwen2.5-3B-Instruct on Re-DocRED. Table 11 re-

ports these results together with the LLaMA3-8B-Instruct and LLaMA3.1-8B-Instruct backbones presented in Section 5.2. We also include LMRC, the strongest LLM-based pipeline baseline in Table 1, as a reference. GLINT maintains competitive performance across all evaluated backbones. Qwen2.5-7B-Instruct trails LLaMA3-8B-Instruct by only 1.88 F1 on the development set and 1.85 F1 on the test set, showing that transferring GLINT to a different model family of comparable scale incurs only a modest degradation. Reducing the Qwen backbone from 7B to 3B lowers test F1 by 1.78 points. Nevertheless, GLINT with Qwen2.5-3B-Instruct still exceeds LMRC by 0.90 test F1 and 0.24 test Ign F1, despite using a substantially smaller backbone. These results suggest that GLINT’s effectiveness is not tied to a single backbone family or parameter scale. Since existing pipeline baselines do not report results with Qwen2.5 backbones, we interpret this experiment as evidence of robustness across backbone families and sizes rather than as a strictly backbone-controlled comparison with LMRC.

## K Prompt Template

Table 12 shows the prompt templates used by the two LLM forward passes in GLINT. Both prompts share a system instruction, the document text, and the entity list, followed by module-specific suffixes that elicit either entity representations or relation predictions. Placeholders of the form  $\{\dots\}$  are substituted at runtime with the corresponding input values. Both templates are zero-shot — no in-context demonstrations are provided, and supervision comes entirely from the QLoRA fine-tuning objective.

| Module                                              | Prompt Template                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Differentiable<br>Candidate-pair<br>Filter<br>(DCF) | <p>You are a relation extraction system. Given a document and a list of entities, identify which entity pairs may have a relation.</p> <p>[Document]<br/>{document_text}</p> <p>[Entities]<br/>E0: {entity_0_name}<br/>E1: {entity_1_name}<br/>...<br/>E{N-1}: {entity_{N-1}_name}</p> <p>E0: {entity_0_name} [ENT]<br/>E1: {entity_1_name} [ENT]<br/>...<br/>E{N-1}: {entity_{N-1}_name} [ENT]</p>                                                                                                                                                                                                 |
| Generative<br>Relation<br>Extractor<br>(GRE)        | <p>You are a relation extraction system. Given a document and a list of entities, predict the relation(s) between each given entity pair. Use only the relations defined below.</p> <p>[Relations]<br/>0: {relation_0_label}<br/>1: {relation_1_label}<br/>...<br/>k: {relation_k_label}</p> <p>[Document]<br/>{document_text}</p> <p>[Entities]<br/>E0: {entity_0_name}<br/>...</p> <p>[Query] What is the relation from E{h} ({entity_h_name}) to E{t} ({entity_t_name})?<br/>[REL]<br/>[Query] What is the relation from E{h'} ({entity_h'_name}) to E{t'} ({entity_t'_name})? [REL]<br/>...</p> |

Table 12: Prompt templates for the two LLM forward passes in GLINT. Both modules use block-diagonal attention to process multiple entities or pair queries in a single forward pass.