DICP: Deep In-Context Prompt for Event Causality Identification

Lin Mu[1], Jun Shen[1], Li Ni[1], Lei Sang[1], Zhize Wu[2], Peiquan Jin[3], Yiwen Zhang[1]*

[1]Anhui University, [2]Hefei University,

[3]University of Science and Technology of China,

{mulin, nili, sanglei, zhangyiwen}@ahu.edu.cn shenjun@stu.ahu.edu.cn

wuzz@hfuu.edu.cn jpq@ustc.edu.cn

Abstract

Event causality identification (ECI) is a challenging task that involves predicting causal relationships between events in text. Existing prompt-learning-based methods typically concatenate in-context examples only at the input layer, this shallow integration limits the model's ability to capture the abstract semantic cues necessary for identifying complex causal relationships. To address this limitation, we propose a novel model called **D**eep **I**n-**C**ontext **P**rompt (**DICP**), which injects in-context examples into the deeper layer of a pre-trained language model (PLM). This strategy enables the model to leverage the hierarchical semantic representations formed in deeper layers, thereby enhancing its capacity to learn high-level causal abstractions. Moreover, DICP introduces a multi-layer prompt injection mechanism, distributing diverse in-context examples across multiple transformer layers. This design allows the model to recognize a broader range of causal patterns and improves its generalization across different contexts. We evaluate the DICP model through extensive experiments on two widely used datasets, demonstrating its significant improvement in ECI performance compared to existing approaches. Furthermore, we explore the impact of varying the number of deep layers on performance, providing valuable insights into the optimal layer configuration for ECI tasks. Code is available at https://github.com/sj1071-cell/DICP.

1 Introduction

Event causality identification (ECI) involves predicting the causal relationships between events mentioned in the text. For instance, consider the sentence S_1 : *"John won the competition because of his practice."* Here, the two event mentions ***practice*** and ***won***, are linked through the explicit cue phrase "because of", which directly signals that

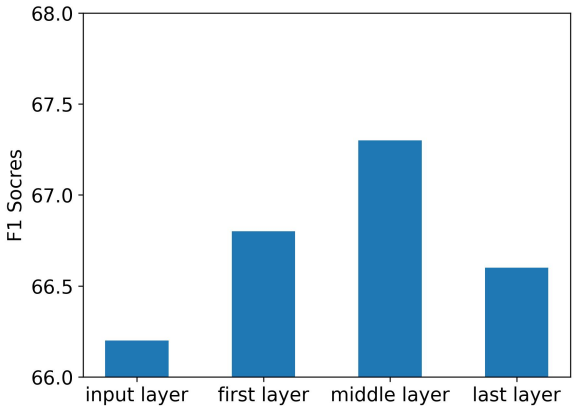

Figure 1: The effect of concatenating prompts in front of different layers used BERT on the ESC dataset.

practice causes ***won***. In contrast, in sentence S_2: *"The reporter was killed during the attack."*, the causal relationship between ***attack*** and ***killed*** is implicit, as it lacks explicit connective phrases. This highlights a key challenge in ECI: While explicit causality may be detected through lexical signals, recognizing implicit causality requires deeper contextual and semantic understanding. Accurate identification of such causal relationships is foundational for a wide range of downstream natural language processing (NLP) applications, such as Question Answering (Oh et al., 2013) and Machine Reading Comprehension (Berant et al., 2014).

Recently, researchers have focused on fine-tuning pre-trained language models (PLMs) to tackle ECI tasks, resulting in significant performance improvements. Among these approaches, the prompt learning paradigm has emerged as particularly promising (Liu et al., 2023b). This approach fine-tunes PLMs by concatenating task-specific prompts with input, guiding the model to generate more accurate predictions. A notable example is the ICCL model (Liang et al., 2024), which incorporates the concept of in-context learning (Dong et al., 2022). ICCL enhances the model's

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 2589–2599
November 4-9, 2025

causal reasoning by concatenating task-specific examples to the input as prompts, allowing the model to infer patterns from relevant prior instances. This method has achieved state-of-the-art performance, particularly in cases involving implicit causality.

However, existing prompt-based ECI models (Liu et al., 2023a; Liang et al., 2024) suffer from a key limitation: they restrict prompt information to the input layer only. This architectural constraint prevents the model from fully leveraging the rich semantic information embedded in task-specific prompts. Recent insights into PLM internals suggest that deeper layers capture more abstract and semantically complex concepts (Skean et al., 2024; Jin et al., 2025). Motivated by this observation, we conducted a preliminary experiment on the ESC (Caselli and Vossen, 2017) dataset, evaluating the effects of injecting in-context examples at various layers of PLMs. As illustrated in Figure 1, injecting in-context examples into deeper layers led to substantial performance improvements over input-level-only in-context examples concatenation. These findings underscore the importance of aligning the semantic complexity of the task with the representational depth of the model.

Building on this insight, we propose a novel model called **D**eep **I**n-**C**ontext **P**rompt (**DICP**) for the ECI task. DICP overcomes the limitations of shallow prompt injection by embedding diverse in-context examples directly into deeper layers of a PLM, thereby enabling the model to capture higher-order causal abstractions. Furthermore, DICP enhances this process by incorporating diverse in-context examples across various deeper layers. This strategy allows the model to efficiently capture diverse causal patterns from these in-context examples, further refining its ability to recognize complex causal relationships. Additionally, we employ an Abstract Meaning Representation (AMR) parser to comprehend semantic structures and extract event-related information from the text.

In summary, our contributions are as follows:

- We propose Deep In-Context Prompt (DICP), a novel prompt learning paradigm for ECI that injects in-context examples into the deeper layer of the PLM. This design enables the model to fully exploit the semantic richness embedded in contextual prompts, particularly for implicit causality.
- We introduce a multi-layer injection strategy that disperses diverse in-context examples across various depths of the PLM. This approach enables the model to effectively capture a wide range of causal patterns, improving both generalization and semantic depth.
- We conduct comprehensive experiments on two widely used ECI datasets. Our results demonstrate that DICP significantly outperforms existing approaches. Furthermore, we analyze the effects of varying the number of injection layers, providing actionable insights into optimal architectural configurations for causality-focused NLP tasks.

2 Related Work

2.1 Tradition ECI Methods

Understanding the causal relationships between events is fundamental for interpreting crucial information in the text. Consequently, event causality identification (ECI) has garnered significant attention from researchers. feature-based and rule-based methods were commonly employed to tackle ECI tasks (Beamer and Girju, 2009; Riaz and Girju, 2014; Ning et al., 2018; Gao et al., 2019). These approaches relied heavily on handcrafted features and predefined rules, limiting their adaptability to diverse contexts and complex relationships. The advent of pre-trained language models (PLMs) revolutionized ECI by enabling models to capture contextual semantic representations and improving their ability to recognize causal relationships. To address the issue of insufficient training data, (Zuo et al., 2020, 2021b) proposed data augmentation methods to generate additional training data. Additionally, (Hu et al., 2023) leveraged an abstract meaning representation (AMR) parser to construct semantic representations of sentences, using semantic structure enhancement to improve recognition capabilities. Although these methods all incorporate PLMs, they adopt the traditional fine-tuning paradigm for classification, which does not leverage the full capabilities of the PLMs.

2.2 Prompt-based ECI Methods

Recently, large-scale PLMs such as GPT (Radford and Narasimhan, 2018), BERT (Devlin et al., 2019), and RoBERTa (Liu, 2019), have achieved significant success in various NLP tasks. However, these PLMs face the issue of a mismatch between the pre-training tasks and downstream tasks, which prevents them from utilizing their full potential. To

address this issue, the prompt-based method was proposed (Brown et al., 2020). It converts downstream tasks into a similar form as pre-training tasks, greatly bridging the gap between the pre-training and fine-tuning stages. Some studies have applied the prompt-based learning paradigm to the ECI task (Liu et al., 2023a; Liang et al., 2024; Wang et al., 2024). For example, (Liu et al., 2023a) used a masked language model and employed a prompt-based paradigm to integrate background and related knowledge from external knowledge bases to predict the answer word at the masked position. (Liang et al., 2024) enhanced the distinction between positive and negative examples by applying contrastive learning within the context, using a PLM to predict causal relationships between event pairs.

2.3 In-Context Learning

The concept of in-context learning (ICL) was first introduced in GPT-3 and mentioned in (Dong et al., 2022). It has since become a widely used method for using large language models (LLMs). ICL is a technique that allows LLMs to learn specific tasks using a small number of labeled examples. The core idea of this method is to design task-related instructions in the form of prompt templates and use a few labeled examples as prompts to guide the model in generating predictions on new test examples.

3 Methodology

To address the challenges of recognizing complex causality patterns in text, we propose the Deep In-Context Prompt (DICP) framework, which integrates three synergistic components: the **Encoding In-Context Module**, the **Deep Prompt Learning Module**, and the **Semantic Structure Module**, as shown in Figure 2. These components are designed to enhance the capabilities of pre-trained language models (PLMs) by incorporating semantic and contextual information tailored for event causality identification (ECI) tasks.

3.1 Problem Definition

We frame the ECI task as a masked prediction problem, where the model infers the causal relation between two events mentioned within a sentence. Given an input instance $x = \{s, e_1, e_2\}$, where s denotes the raw text, and e_1, e_2 represent two event mentions within s, our objective is to predict whether a causal relationship between e_1 and e_2. We introduce two virtual answer words: *Causality* and *NA*, which are added to the vocabulary of the PLM. *Causality* indicates the presence of a causal relationship between the two events, while *NA* indicates the absence of such a relationship. Thus, the output of the PLM denoted as $y \in \{Causality, NA\}$ indicates whether a causal relationship exists between the two events. We then design the prompt template $T(x)$ for input to the PLM:

$$T(x) = s + [start] + e_1 + [MASK] + e_2 + [end].$$

where the tokens $[start]$ and $[end]$ are used to mark the boundaries of the cloze-style template. $[MASK]$ is placed where the model will predict the causal relationship.

3.2 Encoding In-Context Module

Prior methods have struggled to recognize these complex causal patterns. To address this limitation, we propose injecting in-context examples into the deeper layer of the PLM. This approach allows the model to learn from task-specific patterns and improve its ability to identify causal relationships. Since these implicit relationships are difficult to identify, we initially select some sentences with implicit relationships from the training set as a candidate set. Then, we randomly select n examples from the candidate set for relevant sample experiments. The in-context example set is denoted as $Demo = \{d_1, \ldots, d_n\}$, where each in-context example $d_i = \{e_1^i, e_2^i, s^i, y^i\}$ includes the an event mention pair, sentence, and causal label. Each in-context example is then formatted using an in-context example template $T_{icl}(d_i)$:

$$T_{icl}(d_i) = s^i + [start] + e_1^i + y^i + e_2^i + [end].$$

where the tokens $[start]$ and $[end]$ mark the boundaries.

After constructing the in-context example, each is individually encoded using the same BERT model, yielding initial representations for each in-context example, denoted as contextual embedding $H = \{H_1, H_2, \ldots, H_n\}$.

3.3 Deep Prompt Learning Module

Building on the semantic representations of in-context examples captured in Section 3.2, this module integrates these semantic representations into the deep encoder layer of the model. This enables the model to capture nuanced causal patterns at deep levels of abstraction. To facilitate this integration, the prompt template $T(x)$ is converted into

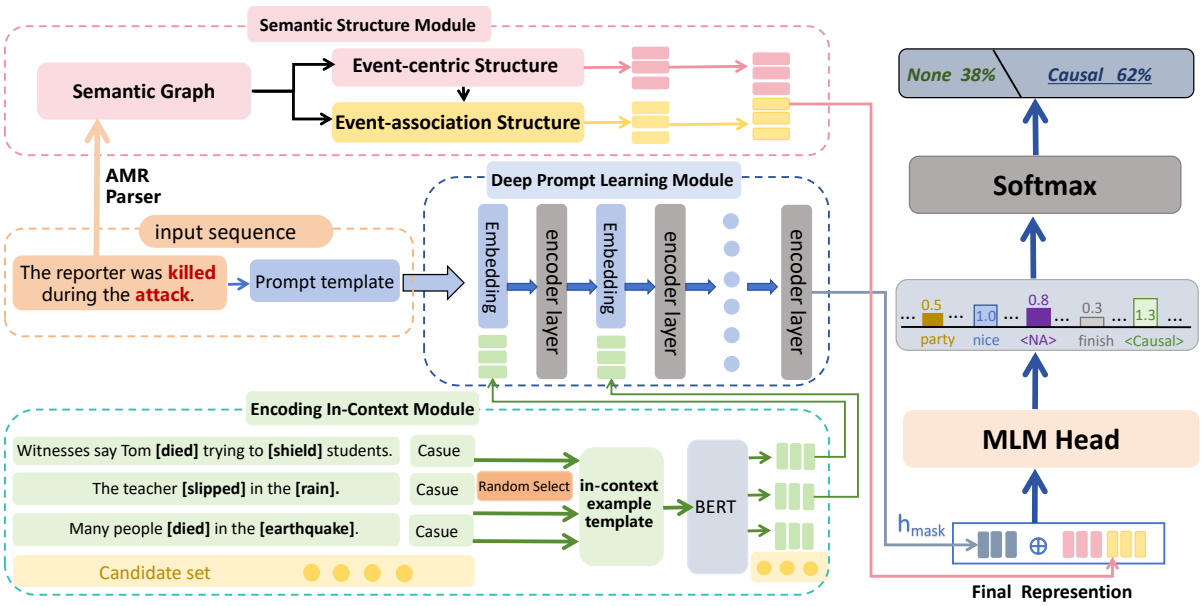

Figure 2: Overview of our DICP framework for ECI. It consists of the Semantic Structure Module (left-upper part), Deep Prompt Learning Module (left-middle part), Encoding In-Context Module (left-lower part), and the Prediction section (right part).

the corresponding embedding representation used by another BERT, denoted as $\varphi((T(x)))$, which serves as the input to the BERT model.

Due to the input length constraints of the BERT model, we inject contextual embedding H into the first n layers of the BERT encoder. This allows the model to integrate the rich diverse causal information from the diverse in-context examples into its initial computations. The input at each layer is the embedding of the example injected into that layer, concatenated with the output from the previous layer. An example is injected into each layer, and the first n layers are considered as deep layers. During training, we use the same H for each input sentence. At the l-th layer, the input embedding is represented as:

$$X_l = Concat(H_l, Z_{l-1}) \tag{1}$$

where $X_l \in \mathbb{R}^{(o_{l-1}+q)*d}$, and q is the length of the H_l, o_{l-1} is the output length of the $l-1$ layer, and d is the embedding dimension; Z_{l-1} denotes the output of the $l-1$ layer and $l <= n$. The output of this layer is denoted as:

$$Z_l = TransformerLayer_l(X_l) \tag{2}$$

When $l > n$, the subsequent layers no longer incorporate contextual embeddings H. The propagation process proceeds as follows:

$$X_l = Z_{l-1} \tag{3}$$

$$Z_l = TransformerLayer_l(X_l) \tag{4}$$

Following the encoding of the final layers, the final representation at the $[MASK]$ position is obtained, denoted as h_{mask}, and captures the causal relationships necessary for accurate prediction.

3.4 Semantic Structure Module

In this section, we describe how to capture explicit semantic relationships encoded in text. The semantic structure module leverages explicit relationships extracted from text to enhance the model's predictions of event causality. This module employs an abstract meaning representation (AMR) (Banarescu et al., 2013) parser to transform the input text into a semantic graph, where vertices represent entities, events, or attributes, and edges represent semantic relationships. From this graph, two complementary semantic structures are extracted: event-centric structure and event-association structure.

The event-centric structure focuses on enriching the representation of event pairs by aggregating information from neighboring nodes and edges in the graph. A Relational graph convolutional network (RGCN) (Schlichtkrull et al., 2018) is applied to the AMR graph to capture these relationships. Given the presence of multiple arguments and their contextual roles, we aggregate semantic information from L-hop neighbors to obtain comprehensive event representations. The final representations of

the events e_1 and e_2 are denoted as h_{e_1} and h_{e_2}, respectively. To mitigate the effect of the relative position, the sum of their embedding as the final representation.

$$F_E^{(e_1,e_2)} = h_{e_1} + h_{e_2} \tag{5}$$

While the event-centric structure focuses on local context, the event-association structure models the semantic path between events. Based on the intuition that shorter paths indicate stronger relationships, we select the shortest path between two events from the AMR graph. Each path is represented as a sequence of nodes and relations, denoted as $(v_1, r_1); (v_2, r_2); ...; (v_m, r_{pad})$, where v_i is the representation of the i-th node, r_i is the representation of the i-th relation edge, and r_{pad} is the special relation added to the last state. The nodes' initial representations are obtained by utilizing the RGCN, while the relation representations are randomly initialized and updated during training. To encode the path, we use a BiLSTM, which generates the final representation of the i-th semantic path:

$$P_i = BiLSTM[(v_1, r_1); ...; (v_m, r_{pad})] \tag{6}$$

When multiple shortest paths exist, an attention mechanism is employed to fuse information across paths. The representation of the event pairs serves as the query, while the representations of the paths act as the keys and values:

$$\alpha_i = \frac{(F_E^{(e_1,e_2)} W_Q)(P_i W_K)^T}{\sqrt{d_k}} \tag{7}$$

$$F_P^{(e_1,e_2)} = \sum_i Softmax(\alpha_i)(P_i W_V) \tag{8}$$

Here α_i represents the attention score of the i-th path, and W_Q, W_V, W_K are learnable parameters.

The final semantic representation integrates both the event-centric and event-association structures. This combined representation is expressed as:

$$F_{Sem} = F_E^{(e_1,e_2)} \oplus F_P^{(e_1,e_2)} \tag{9}$$

3.5 Training and Prediction

The final representation for the [*MASK*] position is formed by concatenating above representations:

$$F_{mask} = h_{mask} \oplus F_{Sem} \tag{10}$$

The prediction and training stages of our DICP model are designed to evaluate and optimize its ability to predict causal relationships effectively. In the prediction stage, we use [*MASK*] token in the input sentence to predict the answer word, indicating the presence or absence of a causal relationship. The final vector representation at the masked position, F_{mask}, is fed into the classifier of the masked language model. This evaluates the probability of each word in the vocabulary V at the [*MASK*] position:

$$P([MASK] = v \in V | T(x)) \tag{11}$$

where $T(x)$ represents the prompt template constructed for the input sentence.

To specifically address causal relationship prediction, we expand the vocabulary by introducing two virtual tokens, *Causality* and *NA*, which represent the presence and absence of a causal relationship, respectively. A $Softmax$ function is then applied to the prediction scores of these two tokens, normalizing their probabilities. For each sample $i \in I$ in the current batch I, there is:

$$P_i(v_i \in V_a | T(x)) = \frac{exp(p_{v_i})}{\sum_{j=1}^{n} exp(p_{v_j})} \tag{12}$$

where V_a = {$Causality$, NA}.

During the training stage, we optimize our model using cross-entropy loss. This loss function minimizes the difference between the predicted label and the true label for each training instance. The cross-entropy loss is defined as follows:

$$L_{DICP} = -\frac{1}{K} \sum_{k=1}^{K} \mathbf{y}^{(k)} log(\hat{\mathbf{y}}^{(k)}) \tag{13}$$

where $\mathbf{y}^k$ and $\hat{\mathbf{y}}^{(k)}$ are the ground-truth label and the predicted label of the k-th training instance, respectively. K represents the number of samples in a training batch.

4 Experiments

4.1 Experiment Setting

Our approach is evaluated on two public datasets: EventStoryLine Corpus version 0.9 (ESC) (Caselli and Vossen, 2017) and Causal-TimeBank (CTB) (Mirza et al., 2014).

ESC contains 22 topics, 258 documents, 4316 sentences, 5334 event mentions, and 1770 causal event pairs. The dataset is divided into a development set consisting of 6 major impactful events and a test set consisting of 16 major impactful events. Same as previous methods (Hu et al., 2023;

Methods	P	R	F1
LSTM (2017)	34.0	41.5	37.4
Seq (2017)	32.7	44.9	37.8
LR+ (2019)	37.0	45.2	40.7
ILP (2019)	37.4	55.8	44.7
KnowDis (2020)	39.7	66.5	49.7
LearnDA (2021b)	42.2	69.8	52.6
CauSeRL (2021a)	41.9	69.0	52.1
GenECI (2022)	59.5	57.1	58.8
KEPT (2023a)	50.0	68.8	57.9
SemSIn (2023)	64.2	65.7	64.9
DFP (2024)	55.9	69.8	62.1
ICCL (2024)	64.9	69.6	67.1
ICCL* (2024)	67.5	73.7	70.4
GPT-3.5-turbo	27.6	80.2	41.0
GPT-4	27.2	**94.7**	42.2
DICP (ours)	70.4	66.2	68.2
DICP (ours)*	**73.5**	69.4	**71.3**

Table 1: Experimental results on ESC. * indicates the model is based on RoBERTa.

Liang et al., 2024), we use the last two topics of ESC as development data and perform 5-fold cross-validation on ESC. To evaluate the performance, we utilize Precision (P), Recall (R), and F1-score (F1) as our metrics.

CTB contains 184 documents, 6813 event mentions, and 318 causal event pairs. Similar to (Liang et al., 2024), we perform 10-fold cross-validation on CTB. Likewise, the performance of Causal-TB is validated using precision (P), recall (R), and F1-score (F1) as the evaluation metrics.

4.2 Baselines

We compare our approach with several baseline models, categorized into feature-based, and PLM-based methods. The detailed introduction of competitors can be found in Appendix A.

Feature-based: For the ESC dataset, we adopted the following baselines: **LSTM** (Cheng and Miyao, 2017), **Seq** (Choubey and Huang, 2017), **LR+ and ILP**, (Gao et al., 2019). For the CTB dataset, our choices are as follows: **RB** (Mirza and Tonelli, 2014), **DD** (Mirza and Tonelli, 2014), **VR-C** (Mirza, 2014).

PLMs-based: DICP is compared to various PLMs-based approaches: **KnowDis** (Zuo et al., 2020), **LearnDA** (Zuo et al., 2021b), **CauSeRL** (Zuo et al., 2021a), **GenECI** (Man et al., 2022), **SemSIn** (Hu et al., 2023), **KEPT** (Liu et al., 2023a), **DFP** (Huang et al., 2024), **ICCL** (Liang et al., 2024)

LLMs: We also compare DICP with large language models (LLMs), including GPT-3.5-turbo *, GPT-4 (Gao et al., 2023).

4.3 Implementation Details

In the experiments, we utilized the pre-trained AMR parser parse_xfm_bart_large v0.1.0. The PLM we used is BERT-base (Devlin et al., 2019), which consists of 12 encoder layers and 12 attention heads, and the dimension of the hidden layer is 768. The learning rate is set to 1e-5. For the RGCN, we also use a learning rate of 1e-5. Additionally, The dimension of added new learnable tokens in the PLM vocabulary is set to 768. The experiments are conducted on a single NVIDIA RTX 3090 GPU, with a batch size of 20, and the number of layers of the deep prompt used in our experiments is 2. When training on the Causal-TB dataset, we follow (Hu et al., 2023) and (Liang et al., 2024) to implement a sampling strategy for positive and negative samples, with sampling rates set to 5 and 0.3, respectively. We apply the AdamW optimization strategy to optimize all models.

4.4 Main Results

The experimental results on the ESC and Causal-TB datasets are shown in Table 1 and Table 2, respectively. Key findings include:

DICP's performance superiority: Our DICP model outperforms other ECI methods in terms of F1 on both datasets, achieving 68.2% on ESC and 61.4% on CTB, respectively. This demonstrates that incorporating in-context examples and injecting them into deeper layers of the model allows for more effective capture of complex causal patterns, significantly improving the performance of the model. This highlights the effectiveness of our approach in improving ECI through a deeper representation learning of causal information.

Comparison with prompt-based methods: The DICP method outperforms a previous prompt-based method, ICCL. This demonstrates the rationality of incorporating different examples into multiple deeper layers, enabling the model to learn distinct information at different layers. Compared to the ICCL method, which concatenates contextual examples with the input sequence and feeds them together into the model to guide the recognition of implicit relationships, DICP effectively

*www.openai.com

Methods	P	R	F1
RB (2014)	36.8	12.3	18.4
DD (2014)	67.3	22.6	33.9
VR-C (2014)	**69.0**	31.5	43.2
KnowDis (2020)	42.3	60.5	49.8
LearnDA (2021b)	41.9	68.0	51.9
CauSeRL (2021a)	43.6	68.1	53.2
GenECI (2022)	60.1	53.3	56.5
KEPT (2023a)	48.2	60.0	53.5
SemSIn (2023)	52.3	65.8	58.3
DFP (2024)	53.7	64.2	58.5
ICCL (2024)	60.5	58.4	59.1
ICCL* (2024)	63.7	68.8	65.4
GPT-3.5-turbo	6.9	82.6	12.8
GPT-4	6.1	**97.4**	11.5
DICP (ours)	56.4	68.6	61.4
DICP (ours)*	62.4	73.5	**67.3**

Table 2: Experimental results on Causal-TB. * indicates the model is based on RoBERTa.

harnesses the capability of PLMs to extract information at deeper levels through the injection of diverse in-context examples across various deeper layers.

Compared with LLMs: Finally, when compared to large language models (LLMs), our DICP method demonstrates superior performance despite its smaller size. Specifically, DICP achieves a 41.5% improvement in F1 over GPT-4 (Gao et al., 2023). The recall of LLMs is high, but the precision is low, indicating that a large number of non-causal event pairs are falsely identified as causal pairs. The main reason for this may be that natural language contains a large number of descriptions of causal relationships, mainly indicated by causal cue words such as "cause" and "therefore". This highlights the importance of fine-tuning for optimizing model performance, especially in specialized tasks like ECI.

4.5 Ablation Study

To evaluate the impact of each component of the DICP model, we conduct ablation experiments on the ESC dataset. The results are summarised in Table 3. In these experiments, the following variants of the DICP model are considered: $w/o.stru$: The model predicts event-causal relationships without using semantic structure information and in-context examples. $w/o.icl$: The model omits in-context examples. $w/o.sem$: The model does not use the semantic structure module. $w/o.deep$: The model concatenates the in-context examples only at the input layer, rather than adding them in the deep layer of the model. $first$: The model concatenates the embedding representations of the examples and then feeds them together as input before the first encoder layer. $second$: The model concatenates the embedding representations of the examples and then feeds them together as input before the second encoder layer. We now discuss the impact of each component on the model's performance:

Methods	P	R	F1	Δ
DICP$_{w/o.stru}$	52.7	66.4	60.8	-
DICP$_{w/o.icl}$	62.6	67.5	64.1	+3.3
DICP$_{w/o.sem}$	67.1	65.6	66.3	+5.5
DICP$_{w/o.deep}$	63.0	68.1	65.3	+4.5
DICP$_{first}$	68.5	65.8	66.7	+5.9
DICP$_{second}$	67.7	66.6	67.1	+6.3
DICP	70.4	66.2	**68.2**	**+7.4**

Table 3: Ablation results on ESC. Δ means the improvement of the F1 relative to DICP$_{w/o.stru}$.

- **Impact of in-context example.** First, we examine the impact of in-context examples, Compared to the full DICP model, DICP$_{w/o.icl}$ show a 4.1% decrease in the F1-score. Comparing DICP$_{w/o.sem}$ with DICP$_{w/o.stru}$, we observe that DICP$_{w/o.sem}$ achieves a 5.5% improvement in the F1-score. This indicates that incorporating in-context information plays a critical role in accurately predicting causal relationships, as it guides the model to focus on relevant patterns in the input.

- **Impact of semantic structure information.** Next, we analyze the role of semantic structure information. DICP$_{w/o.sem}$ demonstrates a 1.9% decrease in the F1 compared to the full DICP model. This demonstrates that semantic structure information is also crucial for the model's ability to identify causal relationships more effectively.

- **Impact of deep layers.** Then, we investigate the effect of deep layers, DICP$_{w/o.deep}$ has a 2.9% decrease in F1 compared to the full DICP model. This decrease highlights the importance of incorporating in-context examples across different deep layers of the model. By adding in-context examples to multiple layers,

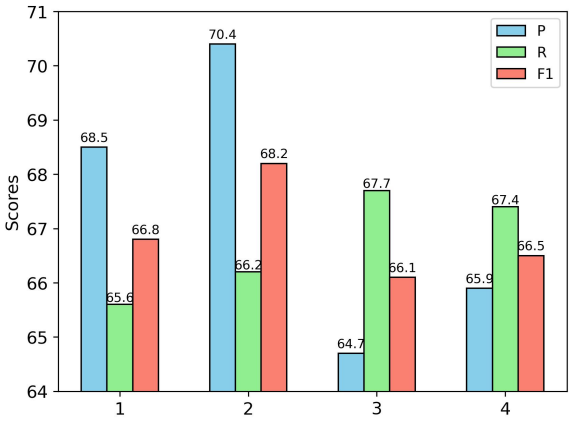

Figure 3: Impact of the number of deep prompt layers on ESC.

the model can better learn and refine complex causal patterns, capturing more causal clues between event pairs.

- **Impact of different layers.** Finally, we investigate the effect of different layers, DICP$_{w/o.deep}$ shows a decrease of 1.4% and 1.8% in F1 compared to DICP$_{first}$ and DICP$_{second}$, respectively. This demonstrates that injecting examples into deeper layers can effectively enhance the model's ability to recognize complex patterns. Compared to the full DICP model, both DICP$_{first}$ and DICP$_{second}$ achieve varying degrees of improvement in F1, indicating that injecting diverse examples into various deeper layers is an effective method to improve prompt-based ECI model.

4.6 Number of Deep Layers

The number of deep layers n is a crucial parameter that influences the model's performance. To assess the impact of n on the model, we conducted experiments using different values of n on the ESC. As shown in Figure 3, the model achieves the best performance when n is set to 2. When $n = 1$, the F1 decreases due to the insufficient number of deep prompt layers, which limits the model's ability to retrieve meaningful information for the ECI task. On the other hand, as n increases beyond 2, the model performance deteriorates significantly. This decline can be attributed to the introduction of excessive in-context examples, which introduce noisy information and interfere with the model's decision-making process. Therefore, the optimal number of deep layers is $n = 2$ balancing the richness of the information and minimizing noise.

Strategy	P	R	F1
Random	54.7	67.8	60.5
Random implicit	56.4	**68.6**	**61.4**
Similar implicit	**58.4**	65.1	61.1
Random implicit false	56.7	66.8	60.6

Table 4: Robustness analysis of the in-context example selection strategy.

4.7 Robustness Analysis

In this section, we explore the robustness of our DICP method under different in-context example selection strategies. To investigate how in-context example selection affects the model, we conducted additional experiments on the CTB dataset using four distinct methods for selecting training examples: 1) **Random**: randomly selecting in-context examples from the training set; 2) **Random implicit**: randomly selecting in-context examples that contain implicit relationships; 3) **Similar implicit**: selecting in-context examples with similar semantic structures. 4) **Random implicit false**: changed the label of one of the two randomly selected implicit examples to an incorrect label.

As shown in Table 4, our DICP method consistently maintains excellent performance across all selection strategies, particularly in terms of F1. This demonstrates the robustness of DICP to varying example selection strategies, highlighting its ability to adapt effectively to different types of data without significant loss in performance.

5 Conclusion

In this paper, we propose DICP for the ECI task. DICP integrates three synergistic components to enhance its performance. First, DICP selects a set of diverse in-context examples and encodes them into semantic embeddings that reflect a wide range of causal scenarios. Second, these embeddings are strategically injected across multiple deeper layers of a PLM, enabling the model to learn abstract, high-level causal representations that go beyond surface-level patterns. Third, DICP incorporates an AMR parser to extract and integrate structured semantic information, thereby further enriching the model's ability to infer complex, implicit causal relationships. We validate DICP through comprehensive experiments on two widely used ECI datasets. Results show that DICP consistently outperforms existing methods, particularly in capturing nuanced and context-dependent causal patterns. By address-

ing the limitations of input-only prompt concatenation, DICP leverages the semantic depth of PLMs to achieve state-of-the-art performance in ECI.

6 Limitations

Our method has the following two limitations: (1) When injecting the selected examples layer by layer, we did not assign different levels of importance to each example. This uniform treatment may have limited the model's ability to effectively leverage the most relevant information from each example. (2) Injecting examples into deeper layers could increase the risk of overfitting, leading the model to rely too heavily on the provided examples while potentially overlooking crucial information in the input text.

7 Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62206004, 62572002, No.62272001, No.62406095), the Natural Science Foundation of Anhui Province (No.2308085MF213), and the Hefei Key Technology RD "Champion-Based Selection" Project (No.2023SGJ011).

References

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Brandon Beamer and Roxana Girju. 2009. Using a bigram event model to predict causal potential. In *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science*, page 430–441.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1510.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv abs/2005.14165*.

Tommaso Caselli and Piek Vossen. 2017. The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the Events and Stories in the News Workshop*, page 77–86.

Fei Cheng and Yusuke Miyao. 2017. Classifying temporal relations by bidirectional lstm over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, page 1–6.

Prafulla Kumar Choubey and Ruihong Huang. 2017. A sequential model for classifying temporal relations between intra-sentence events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, page 1796–1802.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. arXiv preprint arXiv:2301.00234.

Jinglong Gao, Xiao Ding, Bing Qin, and Ting Liu. 2023. Is chatgpt a good causal reasoner? a comprehensive evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, page 11111–11126.

Lei Gao, Prafulla Kumar Choubey, and Ruihong Huang. 2019. Modeling document-level causal structures for event causal relation identification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 1808–1817. Association for Computational Linguistics.

Zhilei Hu, Zixuan Li, Xiaolong Jin, Long Bai, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2023. Semantic structure enhanced event causality identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 10901–10913.

Peixin Huang, Xiang Zhao, Minghao Hu, Zhen Tan, and Weidong Xiao. 2024. Distill, fuse, pre-train: Towards effective event causality identification with commonsense-aware pre-trained model. In *Proceedings of the 2024 Joint International Conference on*

Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), page 5029–5040.

Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. 2025. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 558–573.

Chao Liang, Wei Xiang, and Bang Wang. 2024. In-context contrastive learning for event causality identification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, page 868–881.

Jintao Liu, Zequn Zhang, Zhi Guo, Li Jin, Xiaoyu Li, Kaiwen Wei, and XianSun. 2023a. Kept: Knowledge enhanced prompt tuning for event causality identification. *Knowledge-Based Systems*, 259(110064).

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hieu Man, Minh Nguyen, and Thien Nguyen. 2022. Event causality identification via generation of important context words. In *Proceedings of the 11th Joint Conference on Lexical and Computational Semantics*, page 323–330.

Paramita Mirza. 2014. Extracting temporal and causal relations between events. In *Proceedings of the ACL 2014 Student Research Workshop*, page 10–17.

Paramita Mirza, Rachele Sprugnoli, Sara Tonelli, and Manuela Speranza. 2014. Annotating causality in the tempeval-3 corpus. In *Proceedings of the EACL 2014 Workshop onComputational Approaches to Causality in Language (CAtoCL)*, page 10–19.

Paramita Mirza and Sara Tonelli. 2014. An analysis of causality between events and its relation to temporal information. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 2097–2106.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 2278–2288.

Jong-Hoon Oh, Kentaro Torisawa, Chikara Hashimoto, Motoki Sano, Stijn De Saeger, and Kiyonori Ohtake. 2013. Why-question answering using intra- and inter-sentential causal relations. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1733–1743.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Mehwish Riaz and Roxana Girju. 2014. Recognizing causality in verb-noun pairs via noun and verb semantics. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, page 48–57.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, page 593–607.

Oscar Skean, Md Rifat Arefin, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. In *Workshop on Machine Learning and Compression, NeurIPS*.

Haoyu Wang, Fengze Liu, Jiayao Zhang, Dan Roth, and Kyle Richardson. 2024. Event causality identification with synthetic control. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1725–1737.

Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021a. improving event causality identification via self supervised representation learning on external causal statement. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, page 2162–2172.

Xinyu Zuo, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Weihua Peng, and Yuguang Chen. 2021b. Learnda: Learnable knowledge-guided data augmentation for event causality identification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, page 3558–3571.

Xinyu Zuo, Yubo Chen, Kang Liu, and Jun Zhao. 2020. Knowdis: Knowledge enhanced data augmentation for event causality detection via distant supervision. In *Proceedings of the 28th International Conference on Computational Linguistics*, page 1544–1550.

A Baselines

We compare our approach with several baseline models, categorized into feature-based, and PLM-based methods.

Feature-based: For the ESC dataset, we adopted the following baselines: **LSTM** (Cheng and Miyao, 2017), is a sequential model based on dependency paths; **Seq** (Choubey and Huang, 2017), is a sequential model that utilizes anthropogenic features; **LR+ and ILP**, (Gao et al., 2019) are document structure models; For the CTB dataset, our choices are as follows: **RB** (Mirza and Tonelli, 2014), is a rule-based model; **DD** (Mirza and Tonelli, 2014), is an annotation framework for ECI; **VR-C** (Mirza, 2014), enhances ECI with data filtering and causal signals.

PLMs-based: DICP is compared to various PLMs-based approaches: **KnowDis** (Zuo et al., 2020), is a distant supervision method that enhances data and improves the model using large-scale knowledge bases and external knowledge; **LearnDA** (Zuo et al., 2021b), is a learnable knowledge-guided data augmentation method for ECI; **CauSeRL** (Zuo et al., 2021a), is a method through self-supervised representation learning on external causal statements; **GenECI** (Man et al., 2022), is a generative model through the generation of important context words for ECI; **SemSIn** (Hu et al., 2023), utilizes an AMR parser to obtain semantic graphs, capturing the semantic structure information of the text to enhance identification capability; **KEPT** (Liu et al., 2023a), is a knowledge-enhanced prompt learning paradigm. **DFP** (Huang et al., 2024), used the heterogeneous information fusion module to deeply integrate it with text knowledge, and further enhanced and unified the representation of text and meta-graph with the help of continuous pre-training for ECI. **ICCL** (Liang et al., 2024), introduced in-context contrastive learning modules to improve performance.

LLMs: We also compare DICP with large language models (LLMs), including GPT-3.5-turbo, GPT-4 (Gao et al., 2023), which have demonstrated superior performance across a wide range of tasks due to their extensive pre-training on diverse datasets.