On the Role of Discriminative Models in Generative Relation Extraction

Guozheng Li, Peng Wang*, Zijie Xu, Jing Zhou, Jiajun Liu, Ziyu Shang

School of Computer Science and Engineering, Southeast University
{gzli, pwang, zijiexu, zhoujing0201, jiajliu, ziyus1999}@seu.edu.cn

Abstract

Relation extraction (RE) identifies semantic relations between entities in text, with existing methods falling into two main paradigms: **discriminative** and **generative**. Discriminative models encode sentences and entities into relation representations and classify the most likely relation, whereas generative models directly produce relation labels through sequence generation. Although the latter have benefited from recent advances in large language models (LLMs), their performance remains limited by bottlenecks. In this work, **we present the systematic investigation of how discriminative models can support generative RE**. We propose the **D**iscriminative-to-**G**enerative (**D2G**) framework, which first leverages discriminative models to produce a *top-k* set of candidate relations, and then integrates this knowledge into generative models via in-context or prompt learning. Extensive experiments on five benchmarks demonstrate that D2G consistently achieves state-of-the-art performance, with notable gains on long-tailed relation classes.

1 Introduction

Relation extraction (RE) aims to identify a predefined relation between a given entity pair mentioned in the text. Previous studies (Cabot and Navigli, 2021; Chen et al., 2022a; Li et al., 2023a; Paolini et al., 2021; Sainz et al., 2021; Soares et al., 2019; Yamada et al., 2020; Zhou and Chen, 2022) have achieved remarkable results on various RE datasets by integrating state-of-the-art pre-trained large language models (LLMs) (Devlin et al., 2019; Lewis et al., 2020; Liu et al., 2019; Raffel et al., 2019; Touvron et al., 2023). These approaches generally fine-tune an LLM on downstream datasets and then get the prediction with the highest probability of the output distribution under discriminative or generative paradigms. Discriminative models (Soares et al., 2019; Yamada et al., 2020; Zhou and Chen, 2022) select the relation label with the highest probability based on the output probability distribution of the classification layer, while generative models (Cabot and Navigli, 2021; Paolini et al., 2021) directly generate the relation label name via next token prediction. Both paradigms have been proven to be effective in RE, showing different strengths and weaknesses.

Discriminative methods, typically using LLMs such as RoBERTa (Liu et al., 2019), are trained with masked language modeling to capture fine-grained relation differences, and thus often outperform generative methods in RE (Gutierrez et al., 2022; Li et al., 2023c; Ma et al., 2023). However, directly selecting the highest-probability label can lead to frequent errors. Prior work (Li et al., 2023a) shows that the *top-k* prediction set usually contains the golden label, and leveraging it (e.g., via graph attention (Veličković et al., 2018)) can correct predictions. To analyze this phenomenon, we train a discriminative model (Zhou and Chen, 2022) on SemEval (Hendrickx et al., 2019) and TACRED (Zhang et al., 2017), then obtain *top-k* predictions for each test sample. Results in Figure 1 reveal that (a) labels in the prediction set show high semantic similarity to the golden label (measured by Sentence-BERT (Reimers and Gurevych, 2019)), and (b) recall rapidly increases with k, exceeding 95% at *top*-3 (Li et al., 2023a). These findings highlight that *top-k* prediction sets contain rich **discriminative knowledge**, raising the key question of how to exploit it for improved RE.

Generative methods, based on models such as T5 (Raffel et al., 2019), rely on causal language modeling and generate relation labels through next-token prediction. While LLMs can perform well in downstream tasks via *in-context learning* (ICL) (Brown et al., 2020; Wei et al., 2022; Zhao et al., 2021), open-source generative models often struggle due to limited scale. To address this, *meta in-context learning* (Min et al., 2021b; Chen

*Corresponding author

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 45167–45183

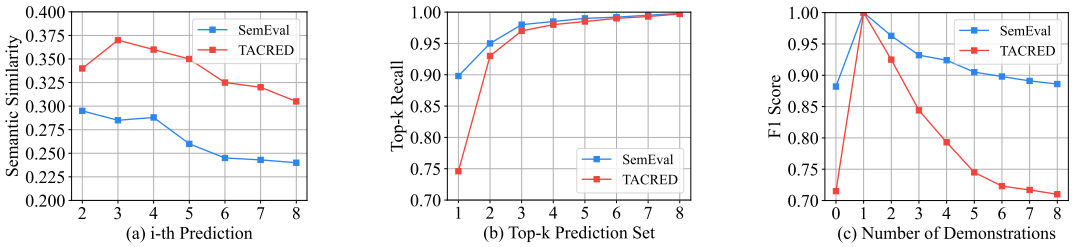

Figure 1: Statistical information of *top-k* prediction set results from a discriminative model, RoBERTa-Large (Liu et al., 2019), and the meta in-context learning results from a generative model, T5-Large (Raffel et al., 2019).

et al., 2022b; Coda-Forno et al., 2024; Li et al., 2024c) fine-tunes models with explicit ICL objectives. We conduct a pilot experiment using T5-Large (Raffel et al., 2019), varying the number of demonstrations ($0 \leq k \leq 8$) that include the golden relation. Results in Figure 1 (c) show that contextual demonstrations improve performance when the golden relation appears and k is moderate. However, ensuring such demonstrations is impractical, and enumerating relations harms accuracy. These limitations motivate us to explore whether discriminative knowledge can effectively enhance generative models.

Motivated by these findings, we propose a new **D**iscriminative-to-**G**enerative (**D2G**) framework that integrates the strengths of both paradigms. Specifically, a discriminative model is trained to produce a *top-k* prediction set for each sample, which contains valuable cues for the golden relation. We further encode these candidate relations into representations and retrieve the most relevant samples, forming **discriminative knowledge**. This knowledge is then injected into a generative model, which is utilized with ICL for proprietary LLMs or fine-tuned with meta ICL for open-source LLMs to effectively leverage it for more accurate and robust predictions. Experiments demonstrate that D2G achieves competitive performance and validates the benefits of combining discriminative and generative approaches. While leveraging discriminative knowledge via ICL is straightforward, it requires retrieval and concatenation with inputs, hindering joint optimization of the two models. Moreover, ICL suffers from *distractors* (Shi et al., 2023), i.e., irrelevant facts in contextual knowledge. To address these issues, we propose injecting discriminative knowledge directly into generative models, which are more robust to distractors when relying on parametric rather than contextual knowledge (Chen et al., 2024; Shi et al., 2023). Concretely, we compress the *top-k* predictions of the discriminative model into a prototype vector and use it as a dynamic prompt. From the prompt learning (PL) perspective, it enhances robustness to distractors and improves the practicality of D2G.

We conduct extensive experiments on five public RE benchmarks, comparing D2G with state-of-the-art methods and fine-tuned baselines. Results show that D2G consistently improves performance across datasets, confirming the effectiveness of leveraging discriminative knowledge to enhance generative models. Notably, PL proves more robust than ICL in handling distractors, enabling generative models to better disentangle useful information. Furthermore, D2G provides substantial gains for long-tailed relations in imbalanced datasets. These findings highlight the broad applicability of our framework: regardless of whether the backbone is a proprietary or open-source model, discriminative knowledge reliably enhances the predictive capability of generative RE models.

2 Method

Notation. We use $f_\theta : x \mapsto f(x, \theta)$ to refer to any parameterized function f with a given set of parameters θ. We describe RE problems using tuples $(s, h, t, r^*, \mathcal{R})$ such that $r^* \in \mathcal{R}$ is the golden relation for the input sentence s given the head entity h and the tail entity t, and use $\mathcal{D}$ to refer to sets of such problems. We drop $\mathcal{R}$ and use only (s, h, t, r^*) when it is clear from context.

Discriminative Models. In a discriminative objective, a parameterized model d_ψ is trained to select the relation label $r \in \mathcal{R}$ with the highest probability of the output distribution. Specifically, we train d_ψ to approximate the probability of r_i given the tuple (s, h, t) using the cross entropy (CE) loss:

$$\mathcal{L}_{\text{CE}}(d_\psi, s, h, t) = -\sum_{i=1}^{|\mathcal{R}|} \mathbb{1}_{r_i = r^*} \log d_\psi(r_i | s, h, t) \quad (1)$$

where $\mathbb{1}_{r_i=r^*}$ is the indicator function which returns 1 if $r_i = r^*$, and $d_\psi(r_i|s,h,t)$ denotes the probability of each relation r approximated by the discriminative model d_ψ.

Generative Models. In a generative objective, a parameterized model g_ϕ is trained to estimate the conditional probability of each token in a sequence given its predecessors: $p(y_l|y_{<l})$. Specifically, we train g_ϕ to approximate p given (s,h,t) using the causal language modeling (CLM) loss:

$$\mathcal{L}_{\text{CLM}}(g_\phi, s, h, t) = -\sum_{l=1}^{L} \log g_\phi(y_l|s,h,t,y_{<l}) \tag{2}$$

where it generates the relation label r^* with length L, and $g_\phi(y_l|s,h,t,y_{<l})$ denotes the probability of each token y in r^* generated by the model g_ϕ.

2.1 In-Context Learning Perspective

To distill the discriminative knowledge into generative models, we introduce the **D2G** from the in-context learning perspective, including (1) *top-k prediction set generating* and (2) *(meta) in-context learning*. Specifically, D2G first generates the *top-k* prediction set $\mathcal{S}$ (i.e., *top-k* relation labels with the highest probabilities) for each sample (s,h,t) using the discriminative model d_ψ. Then for the relations in $\mathcal{S}$, d_ψ retrieves relevant samples in the training corpus as **few-shot demonstrations** (i.e., discriminative knowledge $\mathcal{K}$). Lastly, the generative model g_ϕ uses $\mathcal{K}$ and (s,h,t) as inputs to extract the golden relation r^* via ICL (proprietary LLMs) or meta ICL (open-source LLMs).

Discriminative Knowledge. Generally, a discriminative model d_ψ includes an encoding module $d_{\psi(e)}$ and a classification module $d_{\psi(c)}$. For any arbitrary trained d_ψ, its probability distribution on relation set $\mathcal{R}$ regarding the sample $(s,h,t) \in \mathcal{D}$ is denoted as $\{d_{\psi(c)}(r_i|d_{\psi(e)}(s,h,t)) \,|\, r_i \in \mathcal{R}\}$. We select k relation labels with the highest probability as its *top-k* prediction set $\mathcal{S}$. Then for each relation $r' \in \mathcal{S}$, we need to retrieve a sample (s',h',t',r') from $\mathcal{D}$ that shares similar relation representation with (s,h,t). Specifically, we first encode the relation representation of (s,h,t) as $d_{\psi(e)}(s,h,t)$. Then for each *top-k* relation $r' \in \mathcal{S}$, we select $(s',h',t',r') \in \mathcal{D}$ as the retrieved sample iff $d_{\psi(e)}(s',h',t')$ (i.e. $d^{'}_{\psi(e)}$) is closest to $d_{\psi(e)}(s,h,t)$ (i.e. $d^{*}_{\psi(e)}$) in the hidden space of $d_{\psi(e)}$. Finally, we distill the discriminative knowledge $\mathcal{K}$ w.r.t (s,h,t) from d_ψ and represent it as:

$$\begin{aligned}\mathcal{K} =&\{(s',h',t',r') \,|\, r' \in \mathcal{S}, \forall\,(s'',h'',t'',r') \in \mathcal{D} \\ &\rightarrow |d^{'}_{\psi(e)} - d^{*}_{\psi(e)}| \leq |d^{''}_{\psi(e)} - d^{*}_{\psi(e)}|\}\end{aligned} \tag{3}$$

Note that the architecture of d_ψ is unspecified, which can be any advanced discriminative model.

Algorithm 1 D2G with In-Context Learning

Input: A distribution $p(\mathcal{D})$, a discriminative model d, a generative model g, initial discriminative parameters ψ, initial generative meta-parameters ϕ, learning rate ρ for d, knowledge set size k, and learning rate η for g.
Output: Trained models d_ψ and g_ϕ.

while d is not converged **do**
  $\mathcal{D}^{'} \sim p(\mathcal{D})$
  $\mathcal{L}_{\mathcal{D}'} \leftarrow 0$
  for $(s,h,t) \in \mathcal{D}^{'}$ **do**
    $\mathcal{L}_{\mathcal{D}'} \leftarrow \mathcal{L}_{\mathcal{D}'} + \mathcal{L}_{\text{CE}}(d_\psi, s, h, t)$
  end for
  $\psi \leftarrow \psi - \rho\nabla\frac{1}{|\mathcal{D}'|}\mathcal{L}_{\mathcal{D}'}$
end while
while g is not converged **do**
  $\mathcal{D}^{'} \sim p(\mathcal{D})$
  $\mathcal{L}_{\mathcal{D}'} \leftarrow 0$
  for $(s,h,t) \in \mathcal{D}^{'}$ **do**
    $\mathcal{K} \leftarrow d_\psi(s,h,t,k)$
    $\mathcal{L}_{\mathcal{D}'} \leftarrow \mathcal{L}_{\mathcal{D}'} + \mathcal{L}_{\text{CLM}}(g_\phi, \mathcal{K}, s, h, t)$
  end for
  $\phi \leftarrow \phi - \eta\nabla\frac{1}{|\mathcal{D}'|}\mathcal{L}_{\mathcal{D}'}$
end while
return d_ψ and g_ϕ

Generative Prediction. In general, the discriminative knowledge $\mathcal{K}$ distilled from d_ψ can be utilized by a generative model g_ϕ via (meta) ICL. Specifically, we treat the samples in $\mathcal{K}$ as demonstrations for few-shot prompting, and concatenate $\mathcal{K}$ with (s,h,t) as inputs for extracting relation r^*:

$$r^* = \arg\max_{r' \in \mathcal{S}} g_\phi(r' \,|\, [\mathcal{K}; (s,h,t)]) \tag{4}$$

The generative model g_ϕ is trained and applied to RE by selecting relation $r \in \mathcal{S}$ with the highest probability, based on the next token prediction from the concatenated knowledge $\mathcal{K}$ and sample (s,h,t). Our proposed D2G framework optimizes the following objectives separately:

$$\begin{aligned}\mathcal{L}_{\text{D}} =& \mathbb{E}_{(s,h,t)\sim p(\mathcal{D})}[\mathcal{L}_{\text{CE}}(d_\psi, s, h, t)] \\ \mathcal{L}_{\text{G}} =& \mathbb{E}_{(s,h,t)\sim p(\mathcal{D})}[\mathcal{L}_{\text{CLM}}(g_\phi, \mathcal{K}, s, h, t)]\end{aligned} \tag{5}$$

Note that for proprietary LLMs, $\mathcal{L}_{\text{G}}$ is not available because it directly utilizes $\mathcal{K}$ for in-context prediction. The specific implementation of D2G is not unique because we can customize discriminative and generative models. We provide a simple

D2G baseline implementation in Appendix A. The overall training process is depicted in Algorithm 1.

Model Inference. Given a sample $(s, h, t, r, \mathcal{R})$, we first use the fine-tuned discriminative model d_ψ to obtain the discriminative knowledge $\mathcal{K}$. Then we pass $\mathcal{K}$ and (s, h, t) to the generative model g_ϕ to get the predicted relation r.

2.2 Prompt Learning Perspective

Incorporating discriminative knowledge via ICL is intuitive, but it brings practical challenges: retrieving and formatting examples is cumbersome, and joint optimization with a generative model is infeasible. Moreover, retrieved instances may include distractors that degrade performance (Shi et al., 2023). To overcome these issues, we propose a prompt learning based method. Rather than inserting retrieved examples, we compress the *top-k* predictions from the discriminative model into a prototype vector, which acts as a dynamic prompt for the generative model. This simplifies integration and enables end-to-end training in a unified framework. While prompt-based D2G shares the same high-level pipeline as its ICL-based counterpart, the key difference is in how discriminative knowledge $\mathcal{K}$ is represented. In ICL-based D2G, the *top-k* most similar instances serve as hard labels (all treated equally). In contrast, prompt-based D2G uses a soft-label formulation by encoding the full probability distribution over the *top-k* relations, letting the generative model learn from more nuanced, uncertainty-aware signals.

Algorithm 2 D2G with Prompt Learning

Input: A distribution $p(\mathcal{D})$, a discriminative model d, a generative model g, initial discriminative parameters ψ, initial generative meta-parameters ϕ, learning rate ρ for d, knowledge set size k, and learning rate η for g.
Output: Trained models d_ψ and g_ϕ.

while d or g is not converged **do**
  $\mathcal{D}^{'} \sim p(\mathcal{D})$
  $\mathcal{L}_{\mathcal{D}'} \leftarrow 0$
  for $(s, h, t) \in \mathcal{D}^{'}$ **do**
    $\mathcal{K}_{\text{top-k}} \leftarrow d_\psi(s, h, t, k)$
    $\mathcal{K}_{\text{proto}} = \text{MLP}\left(\mathcal{K}_{\text{top-k}}\,;\,\theta\right)$
    $\mathcal{K}_{\text{prompt}} = \mathcal{V} + \lambda \cdot \mathcal{K}_{\text{proto}}$
    $\mathcal{L}_{\mathcal{D}'} \leftarrow \mathcal{L}_{\mathcal{D}'} + \mathcal{L}_{\text{CE}}(d_\psi, s, h, t) + \mathcal{L}_{\text{CLM}}(g_\phi, \mathcal{K}_{\text{prompt}}, s, h, t)$
  end for
  $\psi \leftarrow \psi - \rho \nabla \frac{1}{|\mathcal{D}'|} \mathcal{L}_{\mathcal{D}'}$
  $\phi \leftarrow \phi - \eta \nabla \frac{1}{|\mathcal{D}'|} \mathcal{L}_{\mathcal{D}'}$
end while
return d_ψ and g_ϕ

Discriminative Knowledge. For the model d_ψ, its probability distribution on relation set $\mathcal{R}$ regarding the sample $(s, h, t) \in \mathcal{D}$ is denoted as $\{d_{\psi(c)}(r_i | d_{\psi(e)}(s, h, t)) \,|\, r_i \in \mathcal{R}\}$. For simplicity, we denote the relation probability distribution computed by the discriminative model as $\mathcal{P} = \{p_1, p_2, ..., p_{|\mathcal{R}|}\}$. We select k relation labels with the highest probability as its *top-k* prediction set $\mathcal{S} = \{(r_i, p_i) \,|\, i \in \text{top-k indices}\}$. We obtain the probability distribution vector $\mathcal{K}_{\text{top-k}} \in \mathbb{R}^k$.

Prompt Generation. We employ a multi-layer perceptron (MLP) to generate the prototype vector:

$$\mathcal{K}_{\text{proto}} = \text{MLP}\left(\mathcal{K}_{\text{top-k}}\,;\,\theta\right) \tag{6}$$

where θ denotes the trainable parameters and $\mathcal{K}_{\text{proto}} \in \mathbb{R}^d$. Next, we integrate $\mathcal{K}_{\text{proto}}$ with the original prompt $\mathcal{V} \in \mathbb{R}^d$ using a residual connection, resulting in the dynamic prompt vector:

$$\mathcal{K}_{\text{prompt}} = \mathcal{V} + \lambda \cdot \mathcal{K}_{\text{proto}} \tag{7}$$

where $\lambda = \sigma(W \cdot [\mathcal{V}; \mathcal{K}_{\text{proto}}])$ is an adaptive weight, learned through a gating mechanism to control the contribution of discriminative signals. σ is the sigmoid function, and $\mathcal{V}$, W are trainable parameters.

Generative Prediction. In general, the discriminative knowledge $\mathcal{K}_{\text{prompt}}$ distilled from d_ψ can be utilized by a generative model g_ϕ via prompt learning. Specifically, we incorporate the vector $\mathcal{K}_{\text{prompt}}$ into the generative model g_ϕ as a prefix prompt, enabling the model to condition its generation on the discriminative knowledge:

$$r^* = \arg\max_{r' \in \mathcal{S}} g_\phi(r' \,|\, [\mathcal{K}_{\text{prompt}}; (s, h, t)]) \tag{8}$$

The discriminative model d_ψ and generative model g_ϕ are jointly trained and applied to RE by selecting relation r with the highest probability. Our proposed D2G framework jointly optimizes the following objective:

$$\begin{aligned}\mathcal{L}_{\text{D2G}} =& \mathbb{E}_{(s,h,t)\sim p(\mathcal{D})}[\mathcal{L}_{\text{CE}}(d_\psi, s, h, t) \\ &+ \mathcal{L}_{\text{CLM}}(g_\phi, \mathcal{K}_{\text{prompt}}, s, h, t)]\end{aligned} \tag{9}$$

Here, $\mathcal{L}_{\text{D2G}}$ contains discriminative loss $\mathcal{L}_{\text{CE}}$ and generative loss $\mathcal{L}_{\text{CLM}}$. The overall training process is depicted in Algorithm 2.

Model Inference. Given a sample $(s, h, t, r, \mathcal{R})$, we first use the fine-tuned discriminative model d_ψ to obtain the discriminative knowledge $\mathcal{K}$. After dynamic prompt generation, we pass $\mathcal{K}_{\text{prompt}}$ and (s, h, t) to the generative model g_ϕ to get the predicted relation r.

Method	Category	Backbone	SemEval	TACRED	TACREV	Re-TACRED	Wiki80
SpanBERT (Joshi et al., 2020)	D	SpanBERT	-	70.8	78.0	85.3	88.1
LUKE (Yamada et al., 2020)	D	LUKE	90.1	72.7	80.6	90.3	89.2
IRE (Zhou and Chen, 2022)	D	RoBERTa	89.8	74.6	83.2	91.1	89.9
KLG (Li et al., 2023a)	D	RoBERTa	90.5	75.6	84.1	-	-
PTR (Han et al., 2022b)	D	RoBERTa	89.9	72.4	81.4	90.9	-
KnowPrompt (Chen et al., 2022a)	D	RoBERTa	90.2	72.4	82.4	91.3	89.0
GenPT (Han et al., 2022a)	D	T5	-	75.3	84.0	91.0	90.6
REBEL (Cabot and Navigli, 2021)	G	BART	89.5	73.7	82.5	-	-
RELA (Li et al., 2023b)	G	BART	90.4	71.2	-	-	-
TANL (Paolini et al., 2021)	G	T5	-	72.1	81.2	90.8	89.1
RE^4 (Li et al., 2024b)	G	T5	90.9	75.6	-	-	-
Discriminative Model (DM)	D	BERT	88.5	72.9	82.2	89.4	88.6
		RoBERTa	89.3	73.9	82.8	90.6	89.5
Generative Model (GM)	G	BART	89.4	73.2	82.1	90.4	89.0
		T5	89.8	73.7	82.7	90.5	89.2
		Llama	90.9	73.9	83.0	90.7	89.8
D2G + ICL (Ours)	D + G	BERT + BART	90.5	75.2	84.0	91.3	90.6
		RoBERTa + T5	90.7	75.5	84.1	91.3	91.1
		RoBERTa + Llama	91.0	75.6	84.0	91.5	91.4
D2G + PL (Ours)	D + G	BERT + BART	91.6	76.3	84.7	92.0	92.0
		RoBERTa + T5	**92.0**	**76.7**	**85.2**	**92.1**	**92.0**
		RoBERTa + Llama	*92.2*	*76.9*	*85.3*	*92.4*	*92.3*

Table 1: Micro-F1 scores of test sets on five RE datasets using open-source models. Results of baselines are retrieved from previous work (Han et al., 2022a; Li et al., 2024b). Previous best results are marked with underline. For fair comparison with previous methods, our best results using RoBERTa and T5 are **bold**, and the best results using larger generative models (Llama) are *italic*.

3 Experiments

3.1 Setup

We experiment our D2G on five RE datasets: **SemEval** (Hendrickx et al., 2019), **TACRED** (Zhang et al., 2017), **TACREV** (Alt et al., 2020), **Re-TACRED** (Stoica et al., 2021), and **Wiki80** (Han et al., 2019). We compare our methods D2G + ICL and D2G + PL against the following baselines: (i) a fine-tuned discriminative model optimized with $\mathcal{L}_{\text{CE}}(d_\psi, s, h, t)$ (**DM**), (ii) a fine-tuned generative model optimized with $\mathcal{L}_{\text{CLM}}(g_\phi, s, h, t)$ (**GM**), and (iii) other state-of-the-art RE models including discriminative fine-tuning methods (**SpanBERT** (Joshi et al., 2020), **LUKE** (Yamada et al., 2020), **IRE** (Zhou and Chen, 2022) and **KLG** (Li et al., 2023a)), discriminative prompt-tuning methods (**PTR** (Han et al., 2022b), **KnowPrompt** (Chen et al., 2022a) and **GenPT** (Han et al., 2022a)), and generative fine-tuning methods (**REBEL** (Cabot and Navigli, 2021), **TANL** (Paolini et al., 2021), **RELA** (Li et al., 2023b) and **RE**^{4} (Li et al., 2024b)). For evaluation, we compute each F1 score from the average across three different runs. For the discriminaitve model, we follow previous works (Chen et al., 2022a; Li et al., 2023a; Zhou and Chen, 2022) and use **RoBERTa-Large** (Liu et al., 2019). For the generative model, we use **T5-Large** (Raffel et al., 2019). To verify the generalization of D2G, we also used **BERT-Large** (Devlin et al., 2019) and **BART-Large** (Lewis et al., 2020) as base models. For larger generative models, we also evaluate `Meta-Llama-3-8B-Instruct` (Grattafiori et al., 2024), `GPT-4o` [1], `DeepSeek-V3-0324` [2] and `DeepSeek-R1-0528` [3]. For proprietary models, we design two methods for demonstration retrieval: (i) **Random + ICL** randomly selects few-shot demonstrations from the training data for each test input, and (ii) **Sentence + ICL** adopts Sentence-BERT (Reimers and Gurevych, 2019) for sentence similarity calculation. For more details on setup, see Appendix B and C.

3.2 Main Results

Open-Source Models. The experimental results in Table 1 demonstrate the consistent effectiveness of D2G across five benchmark datasets using open-source models. When combining RoBERTa as the

[1] https://openai.com/zh-Hans-CN/index/hello-gpt-4o/
[2] https://api-docs.deepseek.com/news/news250325
[3] https://api-docs.deepseek.com/news/news250528

Backbone	Method	SemEval	TACRED	TACREV	Re-TACRED	Wiki80
T5	SFT	89.8	73.7	82.7	90.5	89.2
	D2G + ICL	90.7	75.5	84.1	91.3	91.1
Llama	SFT	90.9	73.9	83.0	90.7	89.8
	D2G + ICL	91.0	**75.6**	**84.0**	**91.5**	**91.4**
GPT-4o	Random + ICL	75.8	37.9	45.2	47.5	26.8
	Sentence + ICL	79.5	38.2	45.8	48.3	29.2
	D2G + ICL	91.4	70.5	76.1	83.4	85.6
DeepSeek-V3	Random + ICL	72.5	35.5	44.8	46.8	28.4
	Sentence + ICL	76.4	38.6	46.2	47.7	29.7
	D2G + ICL	91.9	71.5	78.4	86.7	86.8
DeepSeek-R1	Random + ICL	77.3	38.3	47.4	49.0	31.5
	Sentence + ICL	80.4	39.8	47.5	50.6	32.6
	D2G + ICL	**92.4**	74.0	79.8	87.3	88.1

Table 2: Micro-F1 scores of proprietary models.

DM and T5 as the GM, D2G + PL achieves new state-of-the-art performance on all datasets. These results significantly outperform previous strong discriminative and generative baselines, confirming the advantage of leveraging discriminative knowledge to enhance generative models.

Notably, even when scaling up the generative model to larger architectures such as Llama-3-8B, D2G + PL further improves performance, indicating that the benefits of D2G are preserved and even amplified with more capable generative backbones. Moreover, while base models like BERT and BART individually lag behind state-of-the-art methods, their combination under the D2G framework (e.g., BERT + BART with D2G + PL) achieves highly competitive results, even surpassing many existing SOTA approaches. This underscores the generalizability and model-agnostic nature of D2G, which effectively bridges the gap between discriminative and generative paradigms without requiring specialized architectures or extensive pre-training.

Proprietary Models. Table 2 presents the performance of proprietary LLMs under the D2G + ICL setup, where fine-tuning is not feasible. On datasets with a smaller number of relation types, such as SemEval (9 relations), proprietary models achieve strong results, even outperforming fine-tuned open-source models in some cases. This highlights the strong in-context reasoning ability of large proprietary models in simpler RE settings.

However, on datasets with a larger relation set, such as TACRED (41 relations) and Wiki80 (80 relations), standard ICL methods (Random + ICL and Sentence + ICL) perform poorly, with F1 scores often below 40%. This sharp decline reflects the challenge of disambiguating among a large number of relation labels using only contextual demonstrations. Despite this, D2G + ICL brings substantial improvements, boosting performance by over 30 F1 points in some cases. Although these results still fall short of fully fine-tuned SoTA, they clearly demonstrate that D2G effectively mitigates label confusion in large-relation settings by providing a structured, discriminative prior.

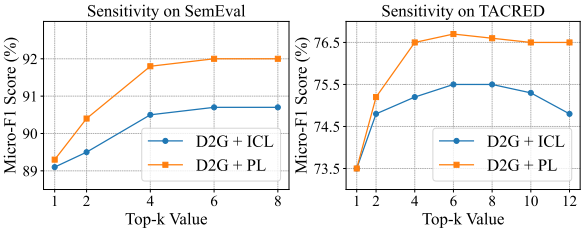

Figure 2: *Top-k* sensitivity results.

These findings affirm that D2G is not only effective for open-source models under fine-tuning settings, but also highly valuable for proprietary models in a black-box ICL scenario. By injecting discriminative knowledge—whether via prompt learning or in-context demonstrations—D2G consistently enhances generative RE, making it more accurate, robust, and scalable across both balanced and long-tailed relation distributions.

4 Analysis

In this section, we delve into the factors that affect D2G and the reasons why this method is effective. For subsequent experiments, we select RoBERTa as the DM and T5 as the GM. For more analysis experiments, refer to Appendix D.

Method	Variant	SemEval	TACRED	TACREV	Re-TACRED	Wiki80
D2G + ICL	Similarity Calculation	**90.7**	**75.5**	**84.1**	**91.3**	**91.1**
D2G + ICL	Random Selection	89.9	75.1	83.6	91.0	90.4
D2G + PL	Full Model	**92.0**	**76.7**	**85.2**	**92.1**	**92.0**
D2G + PL	w/o MLP	91.2	75.8	84.3	91.4	91.3
D2G + PL	w/o Residual	90.9	75.5	84.0	91.0	91.0
D2G + PL	w/o Gating	91.5	76.1	84.7	91.8	91.7

Table 3: Micro-F1 scores of discriminative knowledge and prompt component ablations.

4.1 Sensitivity to *Top-k* Size

We examine how choice of k affects D2G's performance and whether our method is robust to this hyperparameter. We vary k when extracting the *top-k* relation probabilities from the discriminative model on SemEval and TACRED and keep all other training settings fixed. We report the range of F1 scores across k values in Figure 2.

Across both SemEval and TACRED, D2G + ICL and D2G + PL achieve their highest micro-F1 at k = 6, validating our main-experiment setting. In the adjacent range (k= 4–8), scores vary by less than ± 0.5% for ICL and ± 0.7% for PL, underscoring overall robustness. Notably, D2G + PL consistently outperforms the ICL-based variant by roughly 1.0%–1.3% at peak. Furthermore, on TACRED as k increases to 12, PL's soft-label prompts maintain near-peak performance (76.5%), whereas ICL's hard-label approach drops more markedly (to 74.8%), highlighting PL's superior resilience to noisy or irrelevant examples.

4.2 Ablation Studies

For D2G + ICL, we distill the discriminative knowledge from a discriminative model via similarity calculation with its encoding module. To study the necessity of this process, we compare the performance of D2G with discriminative knowledge obtained via similarity calculation and random selection (i.e., randomly select one sample for each *top-k* relation). For D2G + PL, to quantify the contributions of the prototype MLP, the residual fusion, and the gating weight λ, we compare relative drops against the full model to show which component is most critical. We design the following baselines: (i) w/o MLP: Directly use the raw *top-k* probability vector $\mathcal{K}_{\text{top-k}}$ as the prompt (i.e., $\mathcal{K}_{\text{proto}} = \mathcal{K}_{\text{top-k}}$). (ii) w/o Residual: Replace the residual fusion with simple concatenation (i.e. $\mathcal{K}_{\text{prompt}} = [\mathcal{V}; \mathcal{K}_{\text{proto}}]$). (iii) w/o Gating: Fix λ = 1. We report micro-F1 for each variant as shown in Table 3.

Impact of Discriminative Knowledge. Results reveal that similarity-based retrieval consistently outperforms random selection across all datasets (e.g., + 0.8% on SemEval, + 0.4% on TACRED, and + 0.7% on Wiki80). This demonstrates that semantic alignment between the test sample and demonstrations is critical for effective in-context learning. Moreover, random selections introduce noisy or irrelevant examples, degrading the generative model's ability to leverage such knowledge.

Impact of Each Model Component. Removing the residual connection incurs the largest average drop of ± 1.0%, indicating that balancing the original prompt and the prototype vector via a skip-connection is essential for stable integration of discriminative signals. Omitting the MLP leads to an average decrease of ± 0.8%, confirming that learning a compact prototype from the probability distribution extracts richer, more informative cues than using raw scores. Gating weight λ offers modest improvement, showing that while dynamic weighting further refines the prompt, its impact is secondary to the prototype projection and fusion design. Full D2G + PL outperforms all ablated variants by $> 1.0\%$ on average, proving that joint optimization of MLP, residual fusion, and gating is crucial for distilling high-quality discriminative knowledge. Even ablated PL variants (e.g., w/o MLP) generally exceed D2G + ICL (e.g., 91.2 vs. 90.7 on SemEval), reinforcing PL's inherent advantage in mitigating distractors.

4.3 Why does D2G work?

Long-Tail Cases. To uncover why D2G performs so well, we analyzed model errors and found that both discriminative and generative models struggle particularly with long-tailed classes. We report fine-grained results on the test set of TACRED, where we choose 10 relations with the highest frequency as head classes and 10 other relations with the lowest frequency as long-tailed classes. We report F1 scores on these relations, shown in Figure 3.

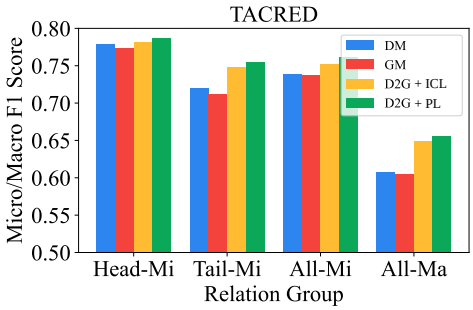

Figure 3: Results of TACRED test set on head classes, tail classes, and all classes. Head-Mi is micro-F1 of head classes. Tail-Mi is micro-F1 of tail classes. All-Mi and All-Ma are micro-F1 and macro-F1 of all classes.

Group	GM	D2G + ICL	D2G + PL	Long-tail (%)
A	84.2	91.8	92.5	8.2
B	63.5	75.1	81.3	45.7
C	32.1	38.4	42.7	63.9

Table 4: Results of the impact of discriminant model error propagation on D2G variants. Long-tail (%) represents the proportion of long-tailed cases in this group.

All models have inferior performances on the long-tailed classes than on the head classes. However, as for the performance gap between these two types of classes, D2G (-3.3%) is much smaller than DM (-5.9%) and GM (-6.2%). Compared with DM and GM, D2G + PL achieves obvious improvements in terms of macro-F1, which shows that D2G is good at handling imbalanced scenarios. Moreover, the improvement of D2G + PL is higher on long-tailed classes than on head classes (i.e., 3.5 and 4.3 point absolute gains for DM and GM, respectively). We attribute this to the usage of D2G framework, while the discriminative knowledge set may contain candidate long-tailed classes that are ignored by previous methods, D2G could revisit these labels and identify the golden long-tailed class. The performance improvements of D2G over previous methods lie in rescuing long-tailed cases.

Error Propagation. When the model's *top-k* predictions omit the correct relation (i.e., the discriminator itself errs), does the D2G framework's performance degrade due to its reliance on incorrect knowledge? In particular, its robustness on long-tail relations should be evaluated. We run the discriminative model on the Re-TACRED test set (whose revised labels are more reliable) and partition the instances into three groups: (i) Group A: the discriminator's *top-1* prediction is correct (i.e., the golden label $\in$ *top-1*). (ii) Group B: the discriminator's *top-1* prediction is incorrect, but the golden label $\in$ *top-k* (with $k = 6$). (iii) Group C: the golden label $\notin$ *top-k* (i.e., the discriminator completely misses the correct relation). We then evaluate each model variant's performance on Group B and C, and compare these results against Group A, as shown in Table 4.

Across these groups, D2G + PL consistently outperforms other variants. Even when the model completely misses the correct label (Group C), PL still surpasses GM by + 10.6% F1. The prototype vector provides auxiliary cues (e.g., similar-relation probabilities) that help the generative model filter out spurious predictions. PL outperforms ICL by + 4.3% F1. This gap underscores that vectorized, soft-label injection is more resilient to irrelevant example noise than concatenating hard-label instances. These results directly demonstrate that D2G + PL does not depend on the discriminative model being perfectly correct: even under complete misclassification, the rich semantic information captured in the *top-k* distribution enhances the generative model's robustness. Moreover, the dramatic gains on Group B create a virtuous feedback loop—since *top-k* already contains the golden label 99% of the time, compressing these signals into a learned prototype reliably boosts long-tail RE.

5 Related Work

Relation extraction (RE) has seen significant advances with transformer-based (Vaswani et al., 2017) LLMs, which can be categorized into discriminative and generative paradigms. Discriminative methods like SpanBERT (Joshi et al., 2020), LUKE (Yamada et al., 2020), and IRE (Zhou and Chen, 2022) learn relation representations for classification, while prompt-based approaches like PTR (Han et al., 2022b), KnowPrompt (Chen et al., 2022a), and GenPT (Han et al., 2022a) design verbalizers for each relation. Generative methods such as REBEL (Cabot and Navigli, 2021), RELA (Li et al., 2023b), and TANL (Paolini et al., 2021) treat RE as a generation task. Recent studies also employ LLMs like Llama (Li et al., 2024b) and GPT-3 (Wan et al., 2023) for RE. Unlike prior work, we systematically explore how discriminative models can enhance generative RE.

LLMs excel in many tasks via in-context learning (ICL) (Chen et al., 2022b; Coda-Forno et al., 2024; Holtzman et al., 2021; Liu et al., 2021; Min

et al., 2021a,b, 2022; Zhao et al., 2021), and recent RE methods (Li et al., 2024a, 2023c; Ma et al., 2023; Wan et al., 2023) use zero (Kojima et al., 2022) and few-shot prompting (Wei et al., 2022) for extraction. While our work also leverages ICL objectives (Min et al., 2021b; Chen et al., 2022b; Coda-Forno et al., 2024) to inject discriminative knowledge, we address ICL's sensitivity to distractors (Shi et al., 2023) by introducing prompt learning (PL) (Ding et al., 2021; Chen et al., 2022a), which encodes knowledge into model parameters for improved robustness.

6 Conclusion

This work demonstrates the significant value of discriminative knowledge for generative RE. Our proposed D2G framework leverages a discriminative model's *top-k* predictions to guide generative inference via in-context or prompt learning, achieving state-of-the-art results across multiple benchmarks. Notably, D2G excels at rescuing long-tailed relations, proving highly effective for imbalanced data. This work pioneers the synergistic integration of discriminative and generative paradigms for RE.

Limitations

While our study demonstrates the effectiveness of incorporating discriminative models into generative relation extraction, there remain several limitations. We mainly evaluate on five standard RE benchmarks with RoBERTa and T5 backbones, and the generality of the approach under different architectures, domains, or low-resource scenarios remains to be further verified. In addition, while we study robustness through controlled noise, real-world deployment may introduce more complex challenges that are not fully captured here.

Acknowledgment

This work was supported by National Science Foundation of China (Grant Nos.62376057) and SEU Innovation Capability Enhancement Plan for Doctoral Students (No. CXJH_SEU 25031).

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of ACL*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *Proceedings of NeurIPS*.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of EMNLP*.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of WWW*.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022b. Meta-learning via language model in-context tuning. In *Proceedings of ACL*.

Zeming Chen, Gail Weiss, Eric Mitchell, Asli Celikyilmaz, and Antoine Bosselut. 2024. Reckoning: reasoning through dynamic knowledge encoding. In *Proceedings of NeurIPS*.

Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matt Botvinick, Jane Wang, and Eric Schulz. 2024. Meta-in-context learning in large language models. In *Proceedings of NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

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*.

Bernal Jimenez Gutierrez, Nikolas McNeal, Clay Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about gpt-3 in-context learning for biomedical ie? think again. In *Findings of EMNLP*.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022a. Generative prompt tuning for relation classification. In *Findings of EMNLP*.

Xu Han, Tianyu Gao, Yuan Yao, Demin Ye, Zhiyuan Liu, and Maosong Sun. 2019. Opennre: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022b. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *Proceedings of EMNLP*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. *arXiv preprint arXiv:1911.10422*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of EMNLP*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *TACL*, 8:64–77.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of NeurIPS*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL*.

Bo Li, Wei Ye, Jinglei Zhang, and Shikun Zhang. 2023a. Reviewing labels: Label graph network with top-k prediction set for relation extraction. In *Proceedings of AAAI*.

Bo Li, Dingyao Yu, Wei Ye, Jinglei Zhang, and Shikun Zhang. 2023b. Sequence generation with label augmentation for relation extraction. In *Proceedings of AAAI*.

Guozheng Li, Wenjun Ke, Peng Wang, Zijie Xu, Ke Ji, Jiajun Liu, Ziyu Shang, and Qiqing Luo. 2024a. Unlocking instructive in-context learning with tabular prompting for relational triple extraction. In *Proceedings of COLING*.

Guozheng Li, Peng Wang, and Wenjun Ke. 2023c. Revisiting large language models as zero-shot relation extractors. In *Findings of EMNLP*.

Guozheng Li, Peng Wang, Wenjun Ke, Yikai Guo, Ke Ji, Ziyu Shang, Jiajun Liu, and Zijie Xu. 2024b. Recall, retrieve and reason: Towards better in-context relation extraction. In *Proceedings of IJCAI*.

Guozheng Li, Peng Wang, Jiajun Liu, Yikai Guo, Ke Ji, Ziyu Shang, and Zijie Xu. 2024c. Meta in-context learning makes large language models better zero and few-shot relation extractors. In *Proceedings of IJCAI*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Yubo Ma, Yixin Cao, YongChing Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of EMNLP*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of ACL*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021b. Metaicl: Learning to learn in context. In *Proceedings of NAACL*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of EMNLP*.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. Structured prediction as translation between augmented natural languages. In *Proceedings of ICLR*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP*.

Oscar Sainz, Oier Lopez de Lacalle, Gorka Labaka, Ander Barrena, and Eneko Agirre. 2021. Label verbalization and entailment for effective zero-and few-shot relation extraction. In *Proceedings of EMNLP*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *Proceedings of ICML*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of AAAI*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of ICLR*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

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 EMNLP*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of NeurIPS*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of EMNLP*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of EMNLP*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of ICML*.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of AACL*.

A Detailed Description of D2G

Discriminative Model Our discriminative model d_ψ is an extension of previous LLM-based RE models (Soares et al., 2019; Zhou and Chen, 2022). Given the input sentence s, we first mark the entity spans h and t using special entity markers (i.e., the modified text s becomes "...@h@...#t#..."), then feed the processed sentence into a discriminative model (e.g. RoBERTa (Liu et al., 2019)) to get its contextual embedding. Finally, we feed the hidden states of the head and tail entities in the last layer obtained by the encoding module $d_{\psi(e)}$, i.e., $\mathbf{h}_{\text{head}}$ and $\mathbf{h}_{\text{tail}}$, into the softmax classifier via the classification module $d_{\psi(c)}$:

$$\begin{aligned} &d_{\psi(e)}(s,h,t) = \text{ReLU}(\mathbf{W}_{\text{proj}}[\mathbf{h}_{\text{head}}, \mathbf{h}_{\text{tail}}]) \\ &d_{\psi(c)}(r \mid d_{\psi(e)}(s,h,t)) \\ &= \frac{\exp(\mathbf{W}_r d_{\psi(e)}(s,h,t) + \mathbf{b}_r)}{\sum_{r' \in \mathcal{R}} \exp(\mathbf{W}_{r'} d_{\psi(e)}(s,h,t) + \mathbf{b}_{r'})} \end{aligned} \tag{10}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{2d \times d}$, $\mathbf{W}_r$, $\mathbf{W}_{r'} \in \mathbb{R}^d$, $\mathbf{b}_r$, $\mathbf{b}_{r'} \in \mathbb{R}^d$ are model parameters. In inference, the classifier returns the relation with the *top-k* maximum probability as the predicted relation set.

Generative Model Our generative model g_ϕ takes the sentence s, head entity h and tail entity t as inputs and directly generates the corresponding relation label name. We concatenate the given sentence and entity pair separated by the recognizable delimiter "|" as the input prompt, and the output is the relation label:

$$|\, s \,|\, h \,|\, t \,| \;\rightarrow\; r \tag{11}$$

To utilize the discriminative knowledge $\mathcal{K}$ via ICL, we treat the samples in $\mathcal{K}$ as demonstrations for few-shot prompting, and concatenate $\mathcal{K}$ with (s,h,t) as inputs to g_ϕ for predicting relation r^*:

$$\begin{aligned} r^* = g_\phi(&|\, s_1 \,|\, h_1 \,|\, t_1 \,|\, r_1 \,|, |\, s_2 \,|\, h_2 \,|\, t_2 \,|\, r_2 \,|, ..., \\ &|\, s_{|\mathcal{K}|} \,|\, h_{|\mathcal{K}|} \,|\, t_{|\mathcal{K}|} \,|\, r_{|\mathcal{K}|} \,|, |\, s \,|\, h \,|\, t \,|\,) \end{aligned} \tag{12}$$

where we provide the specific prompt template and example in Figure 4. In framework optimization, we first train the discriminative model d_ψ. For each sample (s,h,t) in training set, we use the well trained model d_ψ to distill discriminative knowledge $\mathcal{K}$. Then we optimize the generative model g_ϕ with $\mathcal{K}$ and (s,h,t) via fine-tuning with the in-context learning objective.

B Datasets

The statistics of datasets are shown in Table 5. Below we provide the background of each benchmark.

SemEval. The dataset for the SemEval-2010 Task 8 (Hendrickx et al., 2019) is a dataset for multi-way classification of mutually exclusive semantic relations between pairs of nominals. It focuses on semantic relations between pairs of nominals. For example, *tea* and *ginseng* are in an *entity-origin* relation in "The cup contained tea from dried ginseng". It contains 10,717 annotated samples covering 9 relations, covering more abstract relations such as *part-whole*, *cause-effect*, *content-container*, and so on.

TACRED. TACRED (Zhang et al., 2017) is a large-scale RE dataset with 106,264 samples built over newswire and web text from the corpus used in the yearly TAC Knowledge Base Population (TAC KBP) challenges. Samples in TACRED cover 41 relation types as used in the TAC KBP challenges (e.g., *per:schools_attended* and *org:members*) or are labeled as *no_relation* if no defined relation is held. These samples are created by combining available human annotations from the TAC KBP challenges and crowdsourcing.

TACREV. The TACRED-Revisited dataset (Alt et al., 2020) improves the crowd-sourced TACRED dataset (Zhang et al., 2017) for relation extraction by relabeling the dev and test sets using expert linguistic annotators. Relabeling focuses on the 5,000 most challenging samples in dev and test, in total, 51.2% of these are corrected.

Re-TACRED. The Re-TACRED dataset (Stoica et al., 2021) is a significantly improved version of the TACRED dataset (Zhang et al., 2017) for relation extraction. Using new crowd-sourced labels, Re-TACRED prunes poorly annotated sentences and addresses TACRED relation definition ambiguity, ultimately correcting 23.9% of TACRED labels. This dataset contains over 91,000 sentences spread across 40 relations.

Wiki80. Wiki80 (Han et al., 2019) is derived from FewRel (Han et al., 2018), a large scale few-shot dataset. It contains 80 relations and 56,000 instances from Wikipedia and Wikidata (Vrandečić and Krötzsch, 2014). Since Wiki80 is not an official benchmark, we directly report the results on the validation set (Han et al., 2019, 2022a).

INPUT
| I'm not a violent person and the gun was locked in a safe away from anyone. | **gun** | **safe** | **content container**
| The wings of a bat are made of bones like those bones in our arms and hands. | **wings** | **bat** | **component whole**
| A new student organisation is being established at uws from the beginning of 2009. | **student** | **organisation** | **member collection**
| The mist was carried into the air by bursting bubbles over the plating vats. | **mist** | **air** | **entity destination**
| Vegetable stew is a delicious , low calorie and healthy meal to enjoy on a cold winter night. | **vegetable** | **stew** | **entity origin**
| *The factory's workshop functioned inside an extension which was bigger than the actual residence.* | ***factory*** | ***workshop*** |

OUTPUT
component whole

Figure 4: Illustration of the prompt template in generative models. This is an example from the **SemEval** dataset (Hendrickx et al., 2019). The first five demonstrations are retrieved from training corpus by the discriminative model, one of which is the golden relation sample (i.e. *component whole*). The test sample is concatenated with demonstrations and marked with underline. The output of the generative model is simply the relation label name.

Dataset	#Relation	#Train	#Dev	#Test
SemEval	9	6,507	1,493	2,717
TACRED	41	68,124	22,631	15,509
TACREV	41	68,124	22,631	15,509
Re-TACRED	40	58,465	19,584	13,418
Wiki80	80	50,400	5,600	-

Table 5: Statistics of five datasets.

C Implementation Details

We use PyTorch (Paszke et al., 2019) and select RoBERTa-Large (Liu et al., 2019) as the discriminative model and T5-Large (Raffel et al., 2019) as the generative model. All experiments for D2G are conducted on a single NVIDIA A100 (80GB) GPU. All baseline experiments are retrieved from the original papers.

Fine-tuned In-Context Learning. For the discriminative model, the batch size is 16, and the optimizer is AdamW (Loshchilov and Hutter, 2019) with a 1e-5 learning rate ρ and a warm-up strategy. The maximum training epoch is 10, and the maximum input length is 256. We set k as the *top-k* recall achieves close to 1 on the dev set, where we set k as 6, 6, 6, 6 and 5 on SemEval, TACRED, TACREV, Re-TACRED and Wiki80, respectively. For the generative model, we set the batch size to 8 and train the model for 5 epochs with early stopping base on the validation performance. We set the learning rate to 1e-4 and use the AdamW for optimization. We conduct each experiment three times and report the average result to reduce the randomness.

Fine-tuned Prompt Learning. The implementation of discriminative model is the same as that of fine-tuned in-context learning. For the generative model, we set d = 1024. We select the AdamW as the optimizer with a learning rate η of 1e-4. We set the train batch size to 8 due to memory limitations. We train the model for 5 epochs with early stopping and conduct each experiment three times for averaged results. For larger models, we utilize LoRA (Hu et al., 2022) for efficient fine-tuning, and set the rank r of the LoRA parameters to 8 and the merging ratio α to 32. We train Llama for 5 epochs with batch size 4 and learning rate 1e-4. The checkpoint of LoRA adapter that achieves the best result on the validation set is used for testing.

Proprietary models. For three proprietary models GPT-4o, DeepSeek-V3 and DeepSeek-R1, we use the identical prompt construction via API. We construct a prompt for each given test example, which is fed to the model. Each prompt consists of the following components: (i) **Instructions** include a succinct overview of the RE task description and the set of pre-defined relations. The model is explicitly asked to output the relation, which belongs to the pre-defined classes. Otherwise, the model will output `NULL`, (ii) **ICL Demonstrations** are retrieved via our two strategies, and (iii) **Test Input** is concatenated with demonstrations, and LLM is expected to generate the corresponding relation. Specifically, the instruction is *"I will predict the relation between two entities given the context. The pre-defined relations are..."*. Then the demonstration form is shown in Figure 4. For k-shot demonstration, we set k as 6, 6, 6, 6 and 5 on SemEval, TACRED, TACREV, Re-TACRED and Wiki80 for

Discriminator	SemEval	TACRED	TACREV	Re-TACRED	Wiki80
RoBERTa (original)	92.0	76.7	85.2	92.1	92.0
LUKE	92.1	76.8	85.1	92.0	91.9
SpanBERT	91.9	76.6	85.3	92.2	92.1
PTR (prompt-based)	92.0	76.7	85.2	92.1	92.0

Table 6: Effect of replacing the discriminator in D2G + PL. Results show that discriminator choice has negligible impact on final performance.

fair comparison.

Note. In the proprietary scenario, the generative model cannot be fine-tuned, so D2G reduces to: (discriminative model $\rightarrow$ top-k $\rightarrow$ retrieval) + ICL generation. Although the results of DM are not listed in Table 2, they should be the same as the results of DM in Table 1, so we can directly compare the performance differences between the simple DM baseline and ICL. The conclusion is that on datasets with fewer relations, such as SemEval, the performance of proprietary models using ICL is significantly better than that of simple DM baselines. When the dataset has too many relations, such as Wiki80, using ICL for proprietary models will reduce performance due to the inability to fine-tune them. However, the results in Table 2 demonstrate the performance improvement of the D2G+ICL framework on proprietary models compared to the original strategy (Random and Sentence). More absolute performance improvements brought by ICL over SoTA is from the tunable open source models.

D Supplementary Experiments

D.1 Effect of Discriminator Diversity

Experimental Setup. To verify whether the proposed D2G+PL framework heavily depends on the choice of discriminative models, we replace the RoBERTa-based discriminator with several widely-used alternatives: LUKE (Yamada et al., 2020), SpanBERT (Joshi et al., 2020), and a prompt-based discriminative method (PTR (Han et al., 2022b)). All discriminators are fine-tuned under the same hyper-parameter settings as the original RoBERTa backbone. The generator is kept fixed (T5-Large), and the *top-k* predictions from each discriminator are compressed into prototype vectors. Evaluation is conducted on five benchmarks (SemEval, TACRED, TACREV, Re-TACRED, Wiki80). We report micro-F1 scores averaged over three runs with different random seeds.

Analysis. Table 6 presents the results. We observe that replacing the discriminator does not lead to significant differences: the variance across discriminators is within 0.3 F1 points on all datasets. This is because different discriminators tend to agree on the *top-k* predictions, which dominate the construction of prototype vectors. Therefore, the final D2G + PL performance is stable regardless of the discriminator architecture. This indicates that the strength of D2G + PL mainly comes from the prototype integration mechanism rather than the specific discriminator employed.

D.2 Discriminator Noise Injection

Experimental Setup. To further examine the robustness of D2G under imperfect discriminators, we inject controlled noise into the top-k probability distributions produced by the discriminator. Specifically, for each candidate relation, we perturb the logits by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where σ controls the noise level. We then renormalize the probabilities via softmax. We vary $\sigma \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$ and evaluate the final performance of D2G with both ICL and PL strategies. All experiments are conducted on the TACRED dataset with T5-Large as the generator.

Analysis. Figure 5 shows the F1 scores under different noise levels. We observe that D2G + ICL degrades rapidly as the discriminator predictions become noisier, while D2G + PL demonstrates much stronger robustness: even at $\sigma = 0.6$, D2G + PL maintains a performance drop of less than 1.5 points, whereas D2G + ICL suffers a drop of nearly 4 points. These results suggest that the proposed PL variant is considerably more robust to noisy discriminator predictions than the ICL variant. This confirms our intuition that compressing the *top-k* distribution into prototype vectors effectively filters out noise and prevents unstable predictions from propagating into the generator.

D.3 Superficial Ablation Analysis

Experimental Setup. To evaluate the role of discriminative guidance, we conduct two controlled ablations within the same architecture (RoBERTa +

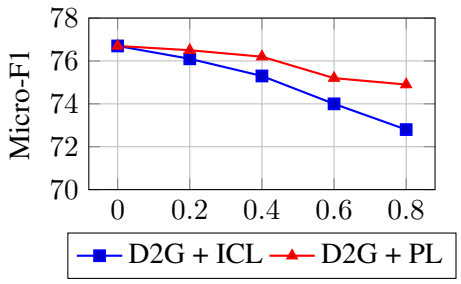

Figure 5: Effect of adding noise to discriminator predictions on TACRED. PL is more robust than ICL under noisy supervision.

T5) and prompt learning framework. (1) **Random top-k vector.** We remove meaningful discriminative signals by replacing the soft top-k probability vector with a randomly sampled vector. Specifically, we sample k values from a uniform distribution $\mathcal{U}(0, 1)$ and normalize them to form a pseudo probability distribution, which is then fed into the prompt module. (2) **Hard top-k vector.** We replace the soft probability distribution with a one-hot vector, where only the highest-probability relation is set to 1 and all others are set to 0. This tests whether fine-grained probability information is necessary. All other components, including model parameters, training procedure, and prompt structure, remain unchanged.

Method (RoBERTa + T5)	SemEval	TACRED
D2G + PL (soft top-k)	92.0	76.7
Random top-k vector	90.2	74.9
Hard top-k (one-hot)	90.9	75.6

Table 7: Ablation study on discriminative guidance.

Analysis. As shown in Table 7, replacing discriminative guidance with random vectors leads to a consistent drop of 1.8–1.9 F1 on both datasets, indicating that the performance gains stem from genuine discriminative knowledge rather than model size or prompt capacity. Moreover, soft top-k probabilities outperform hard one-hot vectors by around +1.1 F1, demonstrating that the full probability distribution provides informative fine-grained cues beyond the top-1 prediction.

D.4 Disconnect between Top-k Recall and Performance Gain.

Experimental Setup. To better understand the relationship between top-k recall and downstream performance, we conduct an additional analysis using the same RoBERTa + T5 framework on TACRED. Instead of focusing on recall, we examine the **entropy of the top-k probability distribution**, which reflects how much uncertainty and fine-grained structure is encoded in the discriminative outputs. We vary k from 3 to 6 and compute both the average entropy of the top-k distribution and the corresponding PL performance.

k	Avg. Entropy	PL Performance
3	0.71	75.9
4	1.02	76.3
5	1.21	76.5
6	1.31	76.7

Table 8: Effect of top-k size on entropy and PL performance on TACRED.

Analysis. We emphasize that top-k recall and PL performance measure fundamentally different properties. Top-k recall is a coarse-grained, binary metric that only evaluates whether the correct label is included in the candidate set. Once the gold label is covered (e.g., recall $\geq 95\%$ at $k = 3$), increasing k does not change this metric. In contrast, PL leverages the **full probability distribution** over the top-k candidates. As shown in Table 8, increasing k leads to higher entropy, indicating richer uncertainty structure. This additional structure provides useful signals beyond simple coverage.

Specifically, larger k: (1) **enriches semantic structure**, as relations often form clusters (e.g., location-related relations), and their relative probabilities encode meaningful similarity patterns; (2) **captures uncertainty**, where distributions such as high $p(r_1)$ with moderate $p(r_2)$ reflect ambiguous cases; (3) **improves hard examples**, where the extra candidates provide additional context for prompt learning. This explains why performance continues to improve even when recall has already saturated at small k.

Relation to the ICL distractor issue. This behavior does not contradict prior findings on distractors in ICL. In ICL, distractors are introduced as *textual examples*, which increase sequence length and may introduce irrelevant semantic content, potentially misleading generation. In contrast, our PL framework operates on **compact vectorized signals**. Here, lower-probability relations do not act as noise but instead contribute structured uncertainty. Therefore, incorrect candidates serve as informative cues rather than harmful distractors.

D.5 Model Size.

Computational cost analysis. We first clarify that the additional computational overhead introduced by the discriminative model (DM) is minimal compared to the generative model (GM). The DM (e.g., RoBERTa) only requires a single forward pass per instance (approximately 12 ms on an A100), while the GM performs autoregressive decoding, which is significantly more expensive. In practice, more than 85–90% of the end-to-end latency is dominated by generation. Therefore, D2G only introduces a modest increase in inference cost while consistently improving performance.

Experimental Setup. We further conduct a size-matched experiment under comparable parameter budgets. Specifically, we compare: (i) RoBERTa-Large (DM-only), (ii) T5-Large (GM-only), and (iii) D2G with RoBERTa-Base (DM) + T5-Base (GM).

Model Setting	SemEval	TACRED	Wiki80
RoBERTa-Large (DM)	89.3	73.9	89.5
T5-Large (GM)	89.8	73.7	89.2
D2G (Base + Base)	88.4	71.8	88.1

Table 9: Size-matched comparison under similar parameter budgets.

Analysis. As shown in Table 9, we observe that combining two base models with D2G does not surpass single large-model baselines. This is expected, as smaller models inherently lack the capacity to capture fine-grained relational semantics. When both the discriminator and generator have limited representational power, the system cannot fully exploit cross-model knowledge transfer. In other words, D2G does not overcome the intrinsic capacity limitations of its backbone models. However, our main results show that D2G with RoBERTa-Large + T5-Large already outperforms a single LLaMA-7B model, despite the latter having significantly more parameters. This demonstrates that the gains of D2G stem from effective integration of discriminative and generative knowledge, rather than simply scaling model size. The goal of D2G is not to outperform arbitrarily larger models with smaller ones. Instead, it shows that, under comparable computational cost to a single generative model, introducing a lightweight discriminative prior yields consistent and meaningful improvements. The discriminative component is highly efficient and contributes structured knowledge at negligible additional cost, enabling better performance without brute-force scaling.

D.6 *k* Selection

Experimental Setup. We further investigate the role of the *top-k* filtering mechanism in our D2G framework. Following our main results, we adopt **RoBERTa-Large** as the discriminator and **T5-Large** as the generator. We set $k = 6$ for SemEval, TACRED, TACREV, and Re-TACRED, and $k = 5$ for Wiki80, consistent with prior experiments. We compare two D2G variants, **ICL** and **PL**, under two inference strategies: (a) *Constrained* decoding (generator restricted to *top-k* labels), and (b) *Adaptive-k* decoding (expanding k until cumulative discriminator probability $\tau \geq 0.9$). For each dataset we report: coverage (% test cases where the gold label is in the *top-k*), F1 when gold $\in$ *top-k* vs. when gold $\notin$ *top-k*, and overall micro-F1 under the three strategies. The results are summarized in Table 10. Constrained F1 (D2G + PL) is aligned with the main results: SemEval (92.0), TACRED (76.7), TACREV (85.2), Re-TACRED (92.1), Wiki80 (92.0).

Analysis. Across all datasets, the gold label is included in the *top-k* discriminator predictions in over 95% of cases, confirming the strong ranking ability of the discriminator. When the gold label is in *top-k*, both ICL and PL achieve strong F1. However, when the gold is missing, performance drops sharply, showing that coverage is critical. Comparing strategies, constrained decoding performs best when coverage is high but fails when coverage is low. After introducing Adaptive-k, there was no significant improvement in consistency compared to the *top-k* method, indicating that the value of k will not have a fundamental impact on the final performance. PL outperforms ICL across all datasets and is less affected by low-coverage cases, confirming its better efficiency-effectiveness trade-off. This analysis demonstrates that our D2G framework is robust to the choice of k as long as gold coverage is high.

D.7 Efficiency Analysis

Experimental Setup. Following the experimental setup, all experiments are run on a single NVIDIA A100 (80GB) GPU with mixed-precision (fp16). We assume a typical relation extraction input length of 128 tokens and an output length of 32

Dataset	Gold in top-k (%)	F1 (gold $\in$ top-k)		F1 (gold $\notin$ top-k)		Constrained F1		Adaptive-k F1	
		ICL	PL	ICL	PL	ICL	PL	ICL	PL
SemEval	98.5	93.0	93.8	36.2	39.4	**90.7**	**92.0**	90.8↑	92.3↑
TACRED	97.2	78.4	79.5	28.7	31.2	**75.5**	**76.7**	75.3↓	76.2↓
TACREV	97.5	86.1	87.0	31.5	34.1	**84.1**	**85.2**	84.0↓	85.0↓
Re-TACRED	98.3	93.0	93.9	33.6	36.5	**91.3**	**92.1**	91.1↓	92.4↑
Wiki80	97.1	91.1	92.0	30.4	33.1	**91.1**	**92.0**	91.4↑	92.2↑

Table 10: *Top-k* analysis (RoBERTa-Large + T5-Large). For each dataset we report coverage (gold in *top-k*), conditional F1 when the gold label is inside / outside the *top-k* (ICL and PL), and overall F1 for Constrained and Adaptive-k decoding. Bolded Constrained F1 entries match the Main Results provided for the paper.

Method	Model(s)	Median Latency (ms / sample)	Throughput (samples/s, batch=8)	Peak Mem. (GB)
Discriminative	RoBERTa-Large	12	600	11
Generative	T5-Large	160	50	23
D2G + ICL	RoBERTa-Large + T5-Large	185	46	37
D2G + PL	RoBERTa-Large + T5-Large	171	58	34

Table 11: Efficiency comparison across discriminative, generative, and hybrid paradigms. Numbers are representative measurements on A100 (80GB) with fp16 precision.

tokens for generative decoding. Inference latency is measured as the median wall-clock time per sample over 200 test examples, including tokenization and GPU synchronization. Throughput is calculated as the number of processed samples per second at batch size 8. Peak GPU memory is obtained from `torch.cuda.max_memory_allocated()` during inference. The results are shown in Table 11.

Analysis.

- **Discriminative vs. Generative:** The discriminative model (encoder-only) is extremely efficient: very low per-sample latency and high throughput, with a modest memory footprint. The generative model is substantially slower due to autoregressive decoding; generation dominates inference time and increases the memory footprint.

- **D2G + ICL vs D2G + PL:** Both D2G variants consistently improve the relation-extraction performance compared to purely discriminative or generative baselines (see main Results section). However, their runtime characteristics differ. **D2G + ICL** requires concatenating k retrieved examples or label-context into the generator input at inference time and may involve an online retrieval/formatting step. This increases encoder cost and memory (longer input sequences) and adds retrieval latency, so the observed per-sample latency and peak memory are the largest among variants. **D2G + PL** uses learned prompt tokens or prototype vectors injected into the generator (no large appended context at runtime). This design keeps the generator input short and avoids retrieval overhead; hence PL attains most of the accuracy gains of D2G while incurring only moderate extra cost relative to the simple hybrid pipeline.

- **Practical implications:** If deployment latency and memory are highly constrained, a discriminative-only model is still preferable. When accuracy is the primary objective, the Generative and D2G variants yield consistent gains; among the three, D2G + PL is a better trade-off for production because it recovers most accuracy improvements with substantially less runtime overhead than D2G + ICL.