Distill-RE: Few-Shot Relation Extraction via Teacher LLM Reasoning Distillation

Anonymous ACL submission

Abstract

Despite their generalization capabilities, large language models (LLMs) have counterintuitively underperformed smaller pre-trained language models (PLMs) on few-shot relation extraction (RE), the task of identifying semantic relationships between entities in text. This gap is primarily due to a systematic over-prediction bias in LLMs. We propose Distill-RE, a two-stage knowledge distillation framework in which a teacher LLM generates chain-of-thought reasoning over existing training examples to supervise two student models: a classification model that predicts relations and a confirmation model that filters false positives. Across student scales from 3B to 70B, Distill-RE establishes a new state-of-the-art among RE models, with its top-performing configuration outperforming the strongest baseline by over 31.5 micro-F1 points averaged across TACRED, TACREV, and Re-TACRED at 8-, 16-, and 32-shot settings. Notably, without synthesizing new relation instances, it outperforms most data augmentation baselines and comes within 1.8 micro-F1 points on average of the strongest.

1 Introduction

Relation extraction (RE) identifies semantic relationships between entities in text, transforming unstructured language into structured knowledge for downstream tasks such as knowledge graph construction, question answering, and semantic search (Zhao et al., 2024; Nayak et al., 2021). Recent advances leveraging pre-trained language models (PLMs) and large language models (LLMs) have achieved strong RE performance, but typically require substantial labeled data (Han et al., 2018; Cabot and Navigli, 2021; Zhao et al., 2024). Annotating RE datasets is particularly challenging due to semantic ambiguity and nuanced relational distinctions that can lead annotators to interpret the same relationship differently, resulting in label noise and low inter-annotator agreement (Hang et al., 2025; Stoica et al., 2021; Alt et al., 2020).

Few-shot RE (FSRE) methods aim to match the performance of fully-supervised models while training on only a small number of labeled examples per relation, directly reducing the annotation burden (Gao et al., 2019). LLM-based few-shot approaches have paradoxically underperformed compared to smaller PLM-based methods (Wan et al., 2023; Xu et al., 2023b; Ma et al., 2023b) due to two factors: (1) limited RE task representation in pre-training data (Wang et al., 2022; Zhang et al., 2023), and (2) systematic over-prediction bias, where models incorrectly assign positive relation labels to instances that should be classified as "no relation" (Wan et al., 2023; Ma et al., 2023b). Since "no relation" comprises roughly 80% of instances in standard benchmarks (Baek and Choi, 2022), this bias has an outsized impact on performance.

We propose Distill-RE, a knowledge distillation RE framework motivated by two insights. First, augmenting labels with teacher-generated chain-of-thought (CoT) explanations is more sample-efficient than learning from labels alone, making reasoning patterns explicit and enabling better generalization from fewer examples. Second, over-prediction bias can be mitigated through task decomposition: rather than training a single model to balance recall and precision, we use specialized classification and confirmation models that optimize for complementary objectives under CoT-augmented supervision.

Our main contributions are:

- We propose Distill-RE, a two-stage knowledge distillation framework using CoT-augmented supervision for few-shot RE.
- Distill-RE sets a new state-of-the-art among RE models, outperforming the strongest baseline by over 31.5 micro-F1 points on average across datasets and shot settings, without rely-

ing on LLM-generated relation instances.

- The confirmation stage reduces over-prediction bias, improving over the classification-only stage by 16.3 macro-F1 and 36.9 micro-F1 points on average across 135 (3 backbone architectures $\times$ 5 seeds $\times$ 3 k values $\times$ 3 datasets) few-shot experimental configurations.
- CoT-augmented supervision consistently improves performance across model sizes and datasets, with mean gains of 9.1 macro-F1 and 7.6 micro-F1 across the same configurations.

2 Background

2.1 Sentence-Level Closed Relation Extraction

Given a sentence s with entity set $\mathcal{E}$ and predefined relation types $\mathcal{R}$, RE produces triplets $\langle e_h, r, e_t \rangle$ where $e_h, e_t \in \mathcal{E}$ are the head and tail entities, $e_h \neq e_t$, and $r \in \mathcal{R}$ is the predicted relation type. Sentence-level RE restricts both entities and evidence to a single sentence, unlike document-level RE where evidence may span multiple sentences. Many RE approaches also leverage entity type information, either as hard constraints (Han et al., 2022a; Yang et al., 2024) or learned features (Cabot and Navigli, 2021), to narrow the relation search space and guide disambiguation. We adopt the former, incorporating entity types directly into our prompt structure and listing only the relations compatible with each entity pair's types, narrowing the candidate relation set.

2.2 Few-Shot Relation Extraction

Few-shot relation extraction aims to achieve strong RE performance with minimal labeled data, typically evaluated under an N-way k-shot setting where N is the number of relation types and k is the number of labeled examples per relation. The primary challenges include: insufficient data for learning discriminative inter-class features, overfitting when fine-tuning with small k, and class imbalance arising from the long-tail distribution of relation types in natural text, where a small number of relations dominate and the majority are rare (Qin et al., 2021; Baek and Choi, 2022).

2.3 The Over-Prediction Problem

A central challenge for LLM-based RE is task-specific resistance to predicting "no relation," as these models exhibit a bias toward assigning a named relation to entity pairs even when none exists (Wan et al., 2023; Ma et al., 2023b). This failure mode is structurally amplified by the RE task formulation. Because many relation types are asymmetric, models must evaluate numerous ordered entity pairs, the vast majority of which do not express any predefined relation (Qin et al., 2021; Baek and Choi, 2022). The resulting extreme class imbalance increases false positives and degrades precision, making reliable "no relation" prediction a central challenge that motivates the two-stage design of our method (Section 4).

3 Related Work

3.1 PLM-Based Few-Shot RE

Prompt-based methods mitigate the mismatch between token-level PLM pre-training objectives and structured RE by reformulating the task as masked language modeling (Chen et al., 2022b; Han et al., 2022b; Zhang et al., 2024) or conditional text generation (Han et al., 2022a; Cabot and Navigli, 2021). Retrieval augmentation has been used to further improve PLM-based label prediction via similarity-weighted neighbor distances (Chen et al., 2022a). Despite these advances, PLM-based approaches fail to leverage the reasoning capacity and sample efficiency that emerge at scale in LLMs (Wei et al., 2022a), reducing their ability to capture fine-grained relation structure with limited data.

3.2 LLM-Based FSRE via In-Context Learning

When applied as frozen backbones under in-context learning (ICL), LLMs are susceptible to prompt-induced biases including majority label and recency effects (Zhao et al., 2021). GPT-RE (Wan et al., 2023) and RAG4RE (Efeoglu and Paschke, 2024) attempt to mitigate prompt sensitivity through improved demonstration retrieval, but both rely on access to the full training set at inference and depart from the strict few-shot setting (Ma et al., 2023a). More fundamentally, because model parameters remain frozen, systematic ICL biases, including positive relation over-prediction, are largely unaddressed.

3.3 LLM-Augmented Data Generation

A complementary research direction uses LLMs purely as data generators to augment the few-shot training set of smaller, fine-tuned PLMs. Unleash (Xu et al., 2023b) and Diverse Samples (Li et al.,

2025) generate schema-constrained synthetic examples to augment PLM classifiers' training sets. ConsistRE (Zheng et al., 2024) incorporates relation-relevant keywords to guide generation and preserve relation consistency, but assumes access to a sufficiently informative external corpus for keyword generation, a requirement that may not hold in practice. ODDA (Zhong et al., 2025) selects training instances with moderate RE model confidence as demonstrations for LLM-based generation, then filters the generated dataset via a global diversity objective. These methods treat the LLM as an offline data source and never directly leverage LLM reasoning for relation classification. Crucially, data generation pipelines are orthogonal to RE model design: they ask how training data can be improved for a fixed model, whereas our method asks how the model can be improved given fixed training data.

3.4 Hybrid LLM–PLM Methods

Beyond offline data generation, several methods incorporate LLM outputs directly into PLM-based training or inference. S^2ynRE (Xu et al., 2023a) fine-tunes an LLM on few-shot gold data to synthesize in-domain training instances, combining synthetic and gold supervision in a two-stage training loop. DSARE (Liu et al., 2024) generates LLM-augmented data to train a PLM-based kNN retrieval model, and uses an LLM to adjudicate ties in the kNN inference stage. TKRE (Guo et al., 2025) generates relational logic explanations conditioned on gold labels to guide PLM pre-training. In all cases, the LLM serves as a data synthesis component and is never fine-tuned to reason over relational semantics. Notably, DSARE leverages the LLM in a zero-shot setting to break ties between down-selected relations.

3.5 LLM Fine-Tuning for RE

In contrast to ICL, fine-tuning updates LLM parameters to internalize task structure, reducing sensitivity to prompt artifacts and alleviating reliance on complex retrieval pipelines (Liu et al., 2022). CoT training (Wei et al., 2022b) extends zero-shot CoT prompting to parameter updates, using teacher-generated explanations to make implicit relational patterns explicit, enabling data-efficient fine-tuning via distillation (Hsieh et al., 2023; Ho et al., 2023). LoRA (Hu et al., 2021) addresses the computational cost of LLM adaptation via low-rank weight updates, achieving strong performance while mitigating overfitting in low-data regimes. LoRA-based fine-tuning for RE has been explored in full-data settings (Efeoglu and Paschke, 2024) but never in the few-shot regime. To our knowledge, no prior work combines parameter-efficient adaptation with reasoning-augmented supervision for FSRE, a gap our method directly addresses.

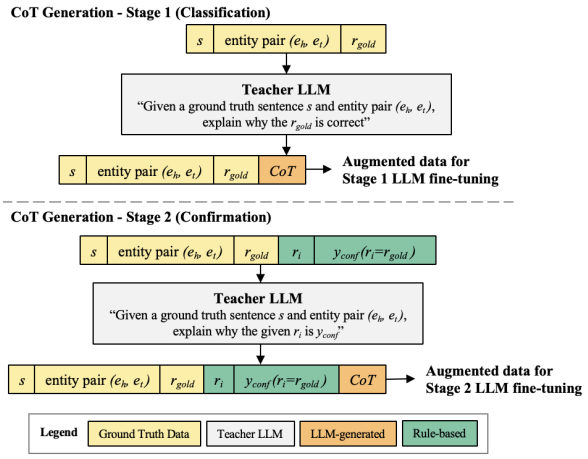

Figure 1: CoT generation pipeline. A zero-shot teacher LLM augments every training sample with a CoT explanation. For Stage 1, the teacher explains why a sample matches its ground-truth relation; for Stage 2, the teacher explains why a candidate r_i, the Stage 1 prediction or a sampled candidate (Section 4.2), is $True$ or $False$. Explanations are paired with labels to fine-tune the student on prediction and reasoning jointly.

4 Methodology

Our approach distills reasoning from a large teacher LLM into student LLMs via a two-stage pipeline: Stage 1, a classification stage that predicts relation r_{pred} for entity pair (e_h, e_t), and Stage 2, a confirmation Stage that validates r_{pred} by predicting $y_{conf} \in \{True, False\}$, filtering false positives and mitigating over-prediction. For each stage, we construct CoT-augmented training data using a frozen teacher LLM, with hard negative sampling in Stage 2. Student LLMs are then fine-tuned on teacher-generated CoT completions and gold labels (relation verbalizations for Stage 1, $True/False$ for Stage 2) via causal language modeling.

4.1 Prompt Construction

All prompts used in our pipeline are provided in full in Appendix B. Teacher prompts were engineered using Mixtral-8x22B (Mixtral-8x22B-Instruct-v0.1) and student prompts using Mistral-7B (Mistral-7B-Instruct-v0.2) with no downstream quantitative optimization.

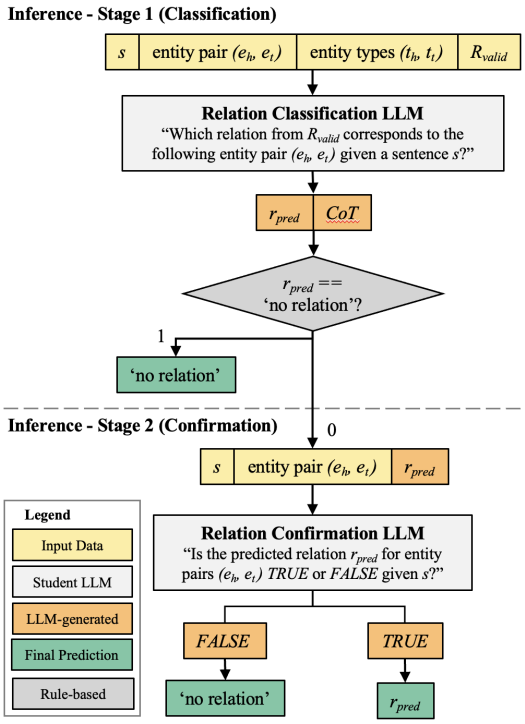

Figure 2: Inference pipeline. Stage 1, the LLM predicts relation r_{pred} with CoT reasoning given input sentence s, entity pair (e_h, e_t) with types (t_h, t_t), and candidate relations $\mathcal{R}_{valid}$. If r_{pred} ="no relation," the pipeline directly outputs "no relation." Stage 2, the LLM verifies r_{pred}. Samples predicted $True$ retain r_{pred}; those predicted $False$ are relabeled as "no relation."

Type-Aware Relation Filtering For Stage 1, we apply type-aware filtering to reduce the candidate relation set. For each entity pair (e_h, e_t) with types (t_h, t_t), only entity type-compatible relations are included in the prompt, producing a filtered set $\mathcal{R}_{valid} \subset \mathcal{R}$ (e.g., "city_of_death" is excluded when both entities are of type "person"). Compatibility is determined using the TACRED family dataset annotation guidelines (Ellis et al., 2015; Stoica et al., 2021), and prior work confirms this filtering improves classification accuracy (Zhang et al., 2023; Yang et al., 2024).

Teacher Prompt Templates We use Mixtral-8x22B (AI, 2024) as the frozen teacher model for all CoT training data augmentation, an open-weight model that achieved state-of-the-art reasoning performance at the time of our experiments. The classification teacher prompt provides entities (e_h, e_t), entity types (t_h, t_t), gold relation r_{gold}, its definition d_{gold}, and instructions to output a supporting explanation of the r_{gold} label. The confirmation teacher prompt provides (e_h, e_t), Stage 1 prediction r_{pred}, its definition d_{pred}, the ground truth label $y_{conf} \in \{True, False\}$, and instructions to output a supporting explanation of y_{conf}.

Student Prompt Templates The classification student prompt provides input sentence s, entity pair (e_h, e_t) with types (t_h, t_t), candidate relations $\mathcal{R}_{valid}$ with definitions, and instructions to predict r_{pred} followed by a supporting explanation. The confirmation student prompt provides input sentence s, entity pair (e_h, e_t), Stage 1 prediction r_{pred}, its definition d_{pred}, and instructions to predict y_{conf} followed by a supporting explanation.

4.2 Training Data Construction

Stage 1: Teacher-Augmented Classification Data Each Stage 1 training sample is augmented with a teacher-generated CoT explanation for why r_{gold} holds for (e_h, e_t) in sentence s.

Stage 2: Confirmation Data from Stage 1 Predictions and Sampling To address over-prediction, we introduce a confirmation stage. After training the Stage 1 LLM, we construct Stage 2 data from its predictions on the training set. For samples where $r_{pred} \neq$ "no relation," correct predictions receive $y_{conf} = True$ and incorrect ones receive $y_{conf} = False$. For samples where $r_{gold} =$ "no relation," we generate false examples by pairing (e_h, e_t) with relations randomly selected from $\mathcal{R}_{valid}$.

To increase difficulty, we augment with hard negatives. For each named-relation (i.e., $r_{pred} \neq$ "no relation") prediction, we sample 2 of the 3 most semantically similar type-compatible relations (selected via BERT-base-uncased (Devlin et al., 2019) embeddings of relation verbalizations) to create additional False training examples. The number of hard negatives was selected via hyperparameter sweep (see Subsection 5.4). Each Stage 2 training sample is constructed around a candidate r_i, drawn either from Stage 1 predictions (r_{pred}) or from the sampled negatives. At inference, Stage 2 is only executed when $r_{pred} \neq$ "no relation"; otherwise the pipeline directly outputs "no relation."

4.3 Student LLM Fine-tuning

We fine-tune separate student LLMs for classification and confirmation on teacher-generated prompt completions and relation labels (r_{gold} for Stage 1, y_{conf} for Stage 2) using standard causal language

modeling (CLM), minimizing:

$$\mathcal{L}_{CLM} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, x) \quad (1)$$

where x is the templated prompt, $y = (y_1, \ldots, y_T)$ is the target sequence (gold label followed by CoT reasoning), and θ are the student model parameters. The classification student generates r_{pred} followed by a supporting explanation; the confirmation student generates $y_{conf} \in \{True, False\}$ followed by reasoning. Unlike prior work applying CoT to RE via in-context learning (Wan et al., 2023) or PLM fine-tuning (Guo et al., 2025), we are, to our knowledge, the first to fine-tune student LLMs on teacher-generated CoT completions for RE, training student LLMs on explicit RE reasoning patterns rather than class label associations alone.

4.4 Answer Parsing

We extract predictions using regular expressions targeting the relation verbalization or $True/False$ decision at the start of each response. For Stage 1, we compute BERT embeddings of the parsed string and select the most similar relation in $\mathcal{R}$:

$$r_{pred} = \arg\max_{r \in \mathcal{R}} \cos(\mathbf{e}_{parsed}, \mathbf{e}_r) \quad (2)$$

where $\mathbf{e}_{parsed}$ is the BERT embedding of the parsed output string and $\mathbf{e}_r$ is the embedding of relation r's verbalization. For Stage 2, we directly parse the $True/False$ token to obtain y_{conf}.

5 Experimental Setup

5.1 LLM Selection

We selected three student models, Llama-3B (Meta-Llama-3.2-3B-Instruct), Mistral-7B (Mistral-7B-Instruct-v0.2), and Llama-70B (Meta-Llama-3-70B-Instruct), to span a range of contemporary instruction-tuned architectures and parameter scales. This selection covers small (3B), medium (7B), and large (70B) capacity models, enabling evaluation of Distill-RE under varying computational constraints. We pair these students with Mixtral-8x22B (Mixtral-8x22B-Instruct-v0.1) as an offline teacher model for chain-of-thought generation, explicitly choosing a teacher architecture distinct from the student families to reduce potential overlap in model inductive biases between teacher and student.

5.2 Datasets

We evaluate on three relation extraction benchmarks derived from the original TACRED (Zhang et al., 2017) (41 relation types, newswire and forum text), TACREV (Alt et al., 2020) (corrected TACRED label errors via expert re-annotation of the 5,000 most challenging dev and test examples), and Re-TACRED (Stoica et al., 2021) (refined relation definitions with 23.9% of examples relabeled for improved consistency). As shown in Table 1, all three datasets exhibit substantial "no relation" class imbalance, which highlights one of the core challenges we address in this work. For more details on the datasets, see Appendix A.

Dataset	Train	Dev	Test	no rel%
TACRED	68,124	22,631	15,509	78.6
TACREV	68,124	22,631	15,509	79.9
Re-TACRED	58,465	19,584	13,418	57.9

Table 1: Dataset statistics. no rel% = proportion of "no relation" instances in the test set.

5.3 Few-shot Sampling

All experiments use a low-resource k-shot setting, where the training and validation sets are each constructed by randomly sampling k examples per relation across all relation types ($N = |\mathcal{R}|$). Final performance is reported on the full test set. We use the sampling code from (Han et al., 2022a) and report results over 5 random seeds, following (Gao et al., 2021).

5.4 Implementation Details

Compute All experiments were implemented in PyTorch (Paszke et al., 2019) using the HuggingFace Transformers library (Wolf et al., 2019). Student models were trained with supervised fine-tuning under a causal language modeling objective using a single NVIDIA H100 NVL GPU. Parameter-efficient fine-tuning was performed using LoRA adapters via the PEFT framework (Mangrulkar et al., 2022). All models were fine-tuned with LoRA adapters in bf16. Llama-3B and Mistral-7B used full bf16 precision, while Llama-70B used QLoRA with the base model loaded in 4-bit quantization to accommodate GPU memory constraints. LoRA was applied to all attention and feed-forward projection layers as well as the language modeling head. Teacher models were hosted with vLLM (Kwon et al., 2023) for efficient GPU

Model	LR	WD	Rank	α	LoRA Params (M)
Llama-3B	5×10^{-5}	0.05	32	0.75r	52
Mistral-7B	1×10^{-4}	0	128	0.5r	340
Llama-70B	5×10^{-5}	0.05	64	0.5r	837

Table 2: Hyperparameter sweep results showing the optimal configuration per student model, selected by validation micro-F1.

inference. Mixtral-8x22B was used exclusively for offline CoT generation as a data augmentation step.

Hyperparameter Sweep We conducted a grid search over LoRA rank $\{16, 32, 64, 128\}$, scaling factor $\{0.25r, 0.5r, 0.75r\}$, learning rate $\{5 \times 10^{-4}, 5 \times 10^{-5}, 5 \times 10^{-6}\}$, and weight decay $\{0, 10^{-5}, 10^{-4}, 10^{-3}\}$. For Stage 2, we tested sampling 0, 1, 2 and 3 hard negatives per Stage 1 named-relation prediction, selecting 2 based on validation performance. To limit leakage, prompt development and hyperparameter tuning were completed exclusively on the $k = 8$, seed 21 Re-TACRED split and the resulting settings were applied across all datasets and experiments. We restricted experiments to configurations in which Stage 1 and Stage 2 used identical hyperparameter settings. Final hyperparameter selections are shown in Table 2.

Sequential Training and Checkpoint Selection All final experimental models were trained for 15 epochs. Training proceeded sequentially: the best validation checkpoint from Stage 1 was selected, and its outputs were used as inputs to Stage 2. The validation checkpoint with the highest validation micro-F1 was used for each stage's final test evaluation.

6 Results and Discussion

6.1 Relation Classification Performance

Table 3 shows that Distill-RE with the Llama-70B backbone achieves the strongest overall performance across all datasets and supervision levels, substantially outperforming all PLM-based and hybrid baselines. Aggregating across datasets, Distill-RE Llama-70B achieves mean micro-$F1$ scores of 82.3 ± 1.2, 84.4 ± 2.3, and 84.9 ± 2.0 at the 8-, 16-, and 32-shot settings respectively, substantially outperforming the strongest baseline at each setting (49.9, 53.2, and 53.9; DSARE for TACRED and TACREV, TKRE for Re-TACRED), with absolute improvements of +32.4, +31.3, and +31.0 points.

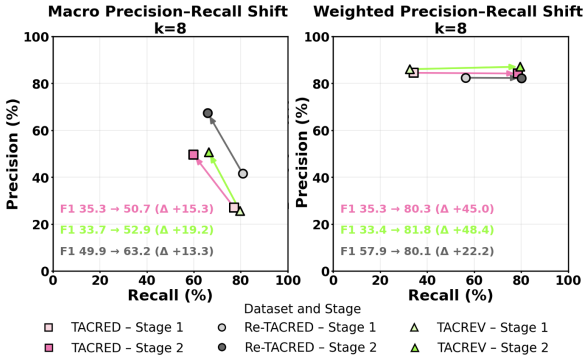

Figure 3: Precision-Recall shift from Stage 1 to Stage 2 of Distill-RE at $k = 8$, averaged across student LLM architectures. Left: macro-averaged; right: weighted-averaged. Arrows indicate the direction and magnitude of the shift, with ΔF_1 annotated per dataset.

Notably, Distill-RE achieves these results without any synthetically generated relation instances. The blue rows in Table 3 report LLM-based data augmentation methods results, which improve existing RE baseline results by augmenting training data with synthetic sentence-relation pairs; a complementary setting not directly comparable to ours. The strongest such method, ODDA, outperforms Distill-RE by 0.4 and 3.2 micro-F1 points at $k = 8$ and $k = 16$ respectively, averaged across TACRED and Re-TACRED (TACREV and $k = 32$ results not reported by the authors), but does so using $8\times k$ additional synthetic samples per run. Whether incorporating synthetic data during Distill-RE training would further improve performance remains an open question, as neither code nor data has been released by these methods. **Within the purely few-shot setting, where only k real labeled examples are available, Distill-RE establishes a new state-of-the-art.**

6.2 Effect of Stage 2 Confirmation Model

To isolate the contribution of the confirmation stage, we compare Stage 1 (classification only) results against their corresponding full two-stage pipeline results. As shown in Figure 3 and Table 4, Stage 1 achieves strong macro-average recall of 77.1%, 79.6%, and 80.8% on TACRED, TACREV, and Re-TACRED respectively at $k = 8$, averaged across three backbone architectures, indicating that the retrieval stage successfully identifies true named-relation instances. Yet Stage 1 weighted recall, dominated by the majority "no relation" class, is substantially lower at 34.2%, 32.5%, and 56.3%, revealing that Stage 1 over-predicts named-relations at the expense of "no relation" classification.

RE Method	Baseline RE Method	Backbone§	TACRED 8	TACRED 16	TACRED 32	TACREV 8	TACREV 16	TACREV 32	Re-TACRED 8	Re-TACRED 16	Re-TACRED 32
LLM-Based RE Methods											
Distill-RE		Llama-3B	75.6 ± 5.7	79.8 ± 3.4	77.2 ± 3.5	78.6 ± 7.1	79.7 ± 3.2	79.7 ± 5.8	77.9 ± 1.2	79.3 ± 1.0	80.6 ± 3.6
Distill-RE		Mistral-7B	78.8 ± 3.5	74.0 ± 4.1	76.6 ± 4.7	77.3 ± 2.1	79.6 ± 5.8	76.2 ± 3.6	79.2 ± 1.6	79.3 ± 1.9	79.6 ± 3.6
Distill-RE		Llama-70B	**81.0 ± 1.9**	**82.8 ± 2.4**	**83.1 ± 4.1**	**82.7 ± 1.2**	**87.0 ± 0.6**	**87.1 ± 0.6**	**83.3 ± 0.8**	**83.5 ± 0.9**	**84.4 ± 0.9**
Baselines: PLM-Based RE Models											
TYP Marker(Zhou and Chen, 2022)†		RoBERTa	*28.9*	*32.0*	*32.4*	*27.6*	*31.2*	*32.0*	*44.8*	*54.1*	*60.0*
PTR(Han et al., 2022b)†		RoBERTa	*28.1*	*30.7*	*32.1*	*28.7*	*31.4*	*32.4*	*51.5*	*56.3*	*62.1*
KnowPrompt(Chen et al., 2022b)†		RoBERTa	*32.0*	*35.4*	*36.5*	*32.1*	*33.1*	*43.7*	*55.3*	*63.3*	*65.0*
GenPT(Han et al., 2022a)		BART	*32.6 ± 0.5*	*35.6 ± 1.8*	*37.1 ± 1.0*	*31.1 ± 0.8*	*34.3 ± 1.9*	*36.6 ± 1.1*	*52.9 ± 1.5*	*58.1 ± 1.8*	*63.9 ± 1.5*
GenPT(Han et al., 2022a)		RoBERTa	*35.7 ± 1.1*	*36.6 ± 1.5*	*37.4 ± 1.8*	*34.4 ± 0.8*	*34.6 ± 1.6*	*36.2 ± 1.8*	*57.1 ± 2.3*	*60.4 ± 1.3*	*65.2 ± 2.0*
GenPT(Han et al., 2022a)		T5	*31.8 ± 0.8*	*33.3 ± 1.3*	*36.0 ± 1.1*	*31.3 ± 1.7*	*32.6 ± 1.5*	*34.6 ± 1.2*	*54.7 ± 2.3*	*58.7 ± 2.4*	*62.5 ± 1.0*
RetrievalRE(Chen et al., 2022a)∗		RoBERTa	*32.0*	*35.2*	*37.1*	*28.9*	*32.3*	*37.3*	*33.7*	*55.8*	*61.8*
PTRE(Zhang et al., 2024)		RoBERTa	*32.7*	*36.5*	*39.2*	*33.9*	*36.3*	*38.1*	*58.8*	*64.4*	*66.3*
Baselines: Hybrid LLM-PLM Models											
S^2ynRE(Xu et al., 2023a) (CP)		RoBERTa	*36.3*	*37.2*	*38.7*	*35.2*	*36.1*	*36.9*	*59.4*	*63.9*	*67.5*
DSARE(Liu et al., 2024)		RoBERTa	***43.8***	***45.4***	***45.9***	***44.7***	***46.6***	***46.9***	*60.0*	*66.8*	*67.1*
TKRE(Guo et al., 2025)	TYP Marker	RoBERTa	*36.7*	*38.3*	*39.1*	*37.2*	*38.2*	*39.0*	*58.7*	*63.8*	*65.5*
TKRE(Guo et al., 2025)	GenPT	RoBERTa	*39.6*	*42.1*	*43.2*	*40.2*	*41.3*	*42.9*	***61.3***	***67.6***	***68.9***
LLM Data Generation Methods											
Unleash(Xu et al., 2023b)	KnowPrompt	RoBERTa / GPT-3.5	*37.9*	*37.4*	-	***42.6***	***41.0***	-	*62.7*	*66.2*	-
ConsistRE(Zheng et al., 2024)‡	TYP Marker	RoBERTa / DeepSeek-v3	*71.8 ± 1.7*	*76.1 ± 1.5*	-	-	-	-	*72.6 ± 3.4*	*81.7 ± 2.5*	-
ConsistRE(Zheng et al., 2024)‡	KnowPrompt	RoBERTa / DeepSeek-v3	*75.8 ± 1.5*	*80.9 ± 0.8*	-	-	-	-	*80.1 ± 1.0*	*84.9 ± 0.6*	-
Diverse(Li et al., 2025) (mix-OBO)	RetrievalRE	RoBERTa Large / LLaMA2-7B-Chat	*33.8*	*35.8*	***37.5***	*33.8*	*34.1*	***36.9***	*58.6*	*62.6*	***65.2***
Diverse(Li et al., 2025) (mix-AAO)	RetrievalRE	RoBERTa Large / LLaMA2-7B-Chat	*34.9*	*36.6*	*37.2*	*35.1*	*35.6*	*36.8*	*55.4*	*61.1*	*64.8*
ODDA(Zhong et al., 2025)	TYP Marker	RoBERTa / DeepSeek-v3	*75.8 ± 1.4*	*80.5 ± 0.7*	-	-	-	-	*76.8 ± 0.9*	*84.9 ± 0.7*	-
ODDA(Zhong et al., 2025)	KnowPrompt	RoBERTa / DeepSeek-v3	***81.0 ± 1.0***	***84.8 ± 0.4***	-	-	-	-	***84.1 ± 0.9***	***87.9 ± 0.3***	-

Table 3: Distill-RE few-shot micro-F_1 results (mean $\pm$ std., higher is better) over five random splits on TACRED, TACREV, and Re-TACRED. Baseline results are reported from their original papers when available; otherwise as reported in (Han et al., 2022a) (†) or (Zhong et al., 2025) (‡). **Red** and **Blue** denote the best and second-best results across the Baseline and Distill-RE sections; **bold** marks the top result per section. Blue rows denote LLM Data Generation (DG) methods, which augment training data with LLM-synthesized samples and use existing methods for RE; these address a complementary setting and are thus not directly comparable. For LLM DG rows, Backbone is formatted as Baseline RE Backbone / DG LLM. Unreported results are marked "–".

Avg	k	Dataset	Stage 1 P	Stage 1 R	Stage 1 F1	Stage 2 P	Stage 2 R	Stage 2 F1
Macro	8	TACRED	27.1	77.1	35.3	49.6	59.8	50.7
		TACREV	25.7	79.6	33.7	50.7	66.3	52.9
		Re-TACRED	41.5	80.8	49.9	67.3	65.9	63.2
Weighted	8	TACRED	84.5	34.2	35.3	84.2	78.6	80.3
		TACREV	86.0	32.5	33.4	87.1	79.5	81.8
		Re-TACRED	82.4	56.3	57.9	82.3	80.1	80.1
Micro	8	TACRED	34.2			78.6		
		TACREV	32.5			79.5		
		Re-TACRED	56.3			80.1		

Table 4: Stage 1 and Stage 2 performance metrics at $k = 8$ averaged across seeds and backbone architectures.

Stage 2 directly addresses the over-prediction problem by reassigning falsely predicted named-relations to the "no relation" class. The confirmation model recovers "no relation" classification while preserving named-relation discrimination, evidenced by weighted recall increases of $+44.4$, $+47.0$, and $+23.8$ percentage points across TACRED, TACREV, and Re-TACRED respectively as weighted precision remains stable. At the macro level, the precision-recall trade-off yields net macro-F1 gains of $+15.4$, $+19.2$, and $+13.3$ points (Table 4).

The impact on micro-F1 is also striking, improving from $(34.2, 32.5, 56.3)$ to $(78.6, 79.5, 80.1)$ on TACRED, TACREV and Re-TACRED respectively. The comparatively smaller gain on Re-TACRED may reflect its cleaner label space, which partially mitigates Stage 1 over-prediction, in combination with its smaller "no relation" percentage, reducing the potential room for improvement by Stage 2. **Taken together, these results validate the two-stage design: named-relation retrieval followed by targeted True/False confirmation effectively balances recall across relation classes with accurate "no relation" rejection.** Consistent trends at $k = 16$ and $k = 32$ are reported in Appendix D.

6.3 Effect of CoT Supervision

To isolate the contribution of incorporating teacher CoT into supervised training, we compare results from matched split-and-seed runs trained with versus without CoT supervision (i.e., gold-label fine-tuning alone). **Our results show that CoT-augmented supervision improves Distill-RE performance across backbone models, shot counts, and datasets, yielding mean gains of** $+7.6$ **micro-F1 and** $+9.1$ **macro-F1 points across 135 runs.** Only 11.1% of runs show degradation, with mean losses of -2.7 micro-F1 and -1.8 macro-F1 points; substantially smaller than typical gains.

Dataset label quality unsurprisingly moderates the effectiveness of CoT-based supervision, as inaccurate reasoning patterns are explicitly distilled into student models. Re-TACRED average im-

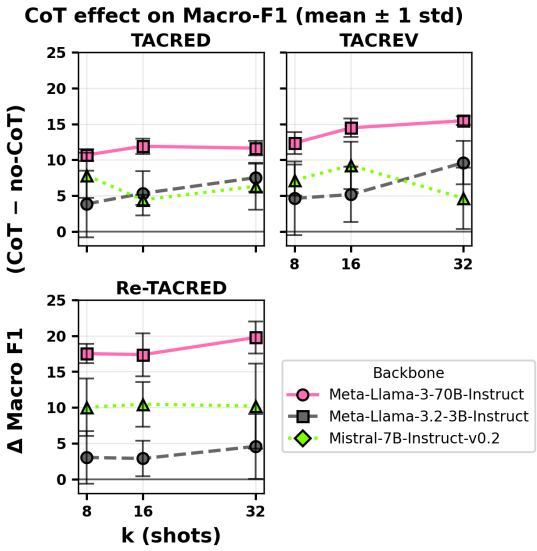

Figure 4: Macro-F1 comparison (CoT − No CoT) between Distill-RE runs with identical k-shot values, backbone model, and random seed across each tested dataset.

provements are largest and monotonically increase with k, suggesting CoT reasoning chains provide reliable signal under relatively clean supervision. With TACRED and TACREV, gains are relatively more modest and exhibit higher variance, consistent with label noise disrupting CoT-based learning. Model capacity further influences robustness to label noise, as evidenced by Llama-70B maintaining the strongest performance gains with the lowest variance across all settings even in the presence of mislabeled data, while the smaller models (Llama-3B, Mistral-7B) show non-monotonic behavior on noisy datasets.

6.4 Runtime Analysis

	Stage 1		Stage 2	
Model	Train	Infer	Train	Infer
Llama-3B	0.1	3.0	0.1	3.6
Mistral-7B	0.3	4.3	0.1	5.0
Llama-70B	1.5	55.3	1.1	40.1

Table 5: Runtime on Re-TACRED (seconds per sample).

Table 5 reports training and inference latency in seconds per sample. Training latency is computed by measuring total fine-tuning time and normalizing by epochs and the number of training samples, using the $k = 8$ setting on Re-TACRED with a maximum sequence length of 1000 tokens (prompt + output). Because TACRED, TACREV, and Re-TACRED share the same sentences and the metric is normalized per sample, latency is expected to be similar across other k values and datasets. Inference latency is averaged over 500 samples using autoregressive generation (`max_new_tokens=500`), which requires multiple forward passes and therefore results in higher latency than training. Runtime increases with model size, with Llama-70B substantially slower than the 3B and 7B models. The increases in model size and runtime reflect the performance-latency trade-off, which practitioners should consider when selecting a configuration for their use case.

7 Conclusion and Future Work

We introduced Distill-RE, a two-stage distillation framework that transfers chain-of-thought reasoning from a teacher LLM to student models for few-shot relation extraction. By decomposing prediction into classification and confirmation, the framework directly addresses over-prediction bias, a central failure mode in RE, and achieves state-of-the-art few-shot performance across TACRED, TACREV, and Re-TACRED among methods that do not leverage LLM-generated relation samples.

Our results highlight two key insights for NLP practitioners:

1. Decoupling recall and precision objectives through a two-stage design leads to substantial and consistent gains, particularly on datasets with severe class imbalance.

2. Reasoning-augmented supervision, rather than labels alone, provides a sample-efficient training signal, enabling strong generalization even in extremely low-resource settings.

For future work, several directions could further strengthen this approach. Integrating Distill-RE with LLM-based data augmentation may provide complementary gains by combining improved supervision with increased data diversity. The quality of teacher-generated reasoning is also a critical factor: exploring stronger teachers, model ensembles, or filtering mechanisms could yield more reliable supervision signals. Finally, extending Distill-RE beyond closed, sentence-level RE to more realistic settings, such as open relation extraction or document-level reasoning, would test the generality of the framework and its ability to scale to more complex information extraction tasks.

Limitations

One of the primary drivers of Distill-RE's few-shot success is also the source of its main limitation, the computationally intensive backbone. LLMs require substantial GPU resources for both training and inference, with costs considerably greater than smaller PLMs, a burden compounded by our finding that performance scales with model size. Distill-RE also depends on a capable teacher model for generating informative relation label explanations. We do not characterize the effect of teacher model quality on overall Distill-RE performance, leaving the teacher model's effect an open question.

Additionally, our algorithm depends on a well-documented dataset schema with clearly defined relations and entity types. We use relation definitions directly in our classification and confirmation prompts, and filter candidate relations based on entity types, meaning incomplete or inconsistent metadata may compromise prompt quality and limit applicability.

Finally, model performance is sensitive to training data quality, as evidenced by our strongest results on Re-TACRED (the cleanest TACRED variant) and weakest on the original, noisiest TACRED. While the few-shot paradigm reduces dependence on large labeled datasets, accurate relation labels remain a key component of our approach. Additionally, our evaluation is limited to TACRED family datasets, which broadly share the same underlying sample sentences and relation schema. Generalization to other domains, relation ontologies and writing styles remains unexamined.

Ethical Considerations

Compared to prior relation extraction approaches, our methodology places particular emphasis on the reasoning capabilities of large language models. However, the pre-training corpora of LLMs are derived predominantly from internet-crawled, human-generated content, and consequently encode the biases of the individuals and societies that produced them. Furthermore, access to the internet is not equally distributed across demographic groups, thus these corpora disproportionately reflect the perspectives of wealthier, Western, and English-speaking populations; referred to by Bender et al. (Bender et al., 2021) as a "hegemonic viewpoint." The biases associated with this viewpoint can produce systematic performance disparities across languages, domains, and perceived gender, sexual identity, race, or ethnicity of the entities whose relations are being modeled (Navigli et al., 2023). This concern is compounded by the fact that relation extraction is commonly used to populate knowledge bases that ground downstream applications, meaning that unmitigated biases will propagate through the entire pipeline. One downstream application worth highlighting specifically is surveillance and monitoring. Relation extraction systems can be used to extract and aggregate structured information about individuals from unstructured text without the knowledge or consent of those being monitored (Brayne, 2017), raising serious concerns regarding the potential for harm caused by a biased upstream RE system.

References

Mistral AI. 2024. Cheaper, better, faster, stronger. https://mistral.ai/news/mixtral-8x22b. Accessed: 2026-02-23.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 1558–1569.

Hyeong-Ryeol Baek and Yong-Suk Choi. 2022. Enhancing targeted minority class prediction in sentence-level relation extraction. *Sensors*, 22(13):4911.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Sarah Brayne. 2017. Big data surveillance: The case of policing. *American sociological review*, 82(5):977–1008.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. Rebel: Relation extraction by end-to-end language generation. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 2370–2381.

Xiang Chen, Lei Li, Ningyu Zhang, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022a. Relation extraction as open-book examination: Retrieval-enhanced prompt tuning. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2443–2448.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022b. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web conference 2022*, pages 2778–2788.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Sefika Efeoglu and Adrian Paschke. 2024. Relation extraction with fine-tuned large language models in retrieval augmented generation frameworks. *arXiv preprint arXiv:2406.14745*.

Joe Ellis, Jeremy Getman, Heather Simpson, Kira Griffitt, Hoa Trang Dang, Ralph Grishman, Heng Ji, Catherine DePrince, Thomas Riese, and Neil Kuster. 2015. TAC KBP 2015 Slot Descriptions. Technical report, Linguistic Data Consortium. Unpublished.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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)*, pages 3816–3830.

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. Fewrel 2.0: Towards more challenging few-shot relation classification. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 6250–6255.

Quanjiang Guo, Jinchuan Zhang, Sijie Wang, Ling Tian, Zhao Kang, Bin Yan, and Weidong Xiao. 2025. Bridging generative and discriminative learning: Few-shot relation extraction via two-stage knowledge-guided pre-training. *arXiv preprint arXiv:2505.12236*.

Jiale Han, Shuai Zhao, Bo Cheng, Shengkun Ma, and Wei Lu. 2022a. Generative prompt tuning for relation classification. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3170–3185.

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 the 2018 conference on empirical methods in natural language processing*, pages 4803–4809.

Tingting Hang, Wei Wu, Jun Feng, Hamza Djigal, and Jun Huang. 2025. A survey of few-shot relation extraction combining meta-learning with prompt learning. *Neurocomputing*, 647:130534.

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large language models are reasoning teachers. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 14852–14882.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Zexuan Li, Hongliang Dai, and Piji Li. 2025. Generating diverse training samples for relation extraction with large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 713–726.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965.

Ye Liu, Kai Zhang, Aoran Gan, Linan Yue, Feng Hu, Qi Liu, and Enhong Chen. 2024. Empowering few-shot relation extraction with the integration of traditional re methods and large language models. In *International Conference on Database Systems for Advanced Applications*, pages 349–359. Springer.

Xilai Ma, Jing Li, and Min Zhang. 2023a. Chain of thought with explicit evidence reasoning for few-shot relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2334–2352.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023b. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the association for computational linguistics: EMNLP 2023*, pages 10572–10601.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: origins, inventory, and discussion. *ACM Journal of Data and Information Quality*, 15(2):1–21.

Tapas Nayak, Navonil Majumder, Pawan Goyal, and Soujanya Poria. 2021. Deep neural approaches to relation triplets extraction: a comprehensive survey. *Cognitive computation*, 13(5):1215–1232.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Yongbin Qin, Weizhe Yang, Kai Wang, Ruizhang Huang, Feng Tian, Shaolin Ao, and Yanping Chen. 2021. Entity relation extraction based on entity indicators. *Symmetry*, 13(4):539.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13843–13850.

Zhen Wan, Fei Cheng, Zhuoyuan Mao, Qianying Liu, Haiyue Song, Jiwei Li, and Sadao Kurohashi. 2023. Gpt-re: In-context learning for relation extraction using large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 3534–3547.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 5085–5109.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022a. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Benfeng Xu, Quan Wang, Yajuan Lyu, Dai Dai, Yongdong Zhang, and Zhendong Mao. 2023a. S2ynre: Two-stage self-training with synthetic data for low-resource relation extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8186–8207.

Xin Xu, Yuqi Zhu, Xiaohan Wang, and Ningyu Zhang. 2023b. How to unleash the power of large language models for few-shot relation extraction? In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 190–200.

Hangxiao Yang, Changpu Chen, Shaokai Zhang, Baiyang Chen, Chang Liu, and Qilin Li. 2024. Etcgn: entity type-constrained graph networks for document-level relation extraction. *International Journal of Machine Learning and Cybernetics*, 15(12):5949–5962.

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. In *Findings of the association for computational linguistics: ACL 2023*, pages 794–812.

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 the 2017 conference on empirical methods in natural language processing*, pages 35–45.

Zirui Zhang, Yiyu Yang, and Benhui Chen. 2024. Prompt tuning for few-shot relation extraction via modeling global and local graphs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13233–13242.

Xiaoyan Zhao, Yang Deng, Min Yang, Lingzhi Wang, Rui Zhang, Hong Cheng, Wai Lam, Ying Shen, and Ruifeng Xu. 2024. A comprehensive survey on relation extraction: Recent advances and new frontiers. *ACM Computing Surveys*, 56(11):1–39.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. Pmlr.

Yifan Zheng, Wenjun Ke, Qi Liu, Yuting Yang, Ruizhuo Zhao, Dacheng Feng, Jianwei Zhang, and Zhi Fang. 2024. Making llms as fine-grained relation extraction data augmentor. In *IJCAI*, pages 6660–6668.

Yijie Zhong, Yunfan Gao, Xiaolian Zhang, and Haofen Wang. 2025. Odda: An ooda-driven diverse data augmentation framework for low-resource relation extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 267–285.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific*

Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 161–168.

A Dataset Details

TACRED The TAC Relation Extraction Dataset (Zhang et al., 2017) is a widely used large-scale dataset for sentence-level relation extraction. It contains sentences from both newswire and discussion forums, and includes 41 relation types. The dataset relations were crowd sourced using Mechanical Turk.

TACREV TACREV (Alt et al., 2020) is a revised version of TACRED that addresses substantial label errors in the original crowdsourced annotations. The authors identified the 5,000 most challenging examples from the development and test sets based on disagreement across 49 relation extraction models, then reannotated them using trained linguists following a rigorous protocol. This process relabeled approximately 50% of the challenging examples while maintaining TACRED's original relation schema and structure.

Re-TACRED Re-TACRED (Stoica et al., 2021) is a reannotated version of the TACRED dataset, created for improved relation label accuracy. The authors refined ambiguous relation definitions to reduce annotator confusion and improve label consistency. The relation labeling process featured an improved crowd-sourced annotation strategy and relabeled 23.9% of the original TACRED dataset.

B Prompt Templates

This section presents all prompts used in the Distill-RE pipeline. Prompts labeled "Teacher" were used to generate teacher model CoT augmentations, while those labeled "Student" were used to prompt student models during fine-tuning and inference. Variable prompt components are denoted by curly brackets { } containing descriptive variable names indicating which text will be inserted into each slot. The rules to fill these variable slots and the role the prompts fill in the Distill-RE pipeline are detailed in Section 4. Variables used across the prompts are defined as follows:

- $sentence$: the sample sentence from which the relation triple is extracted
- $head_entity$, $tail_entity$: the head (subject) and tail (object) entities of interest contained in $sentence$
- $head_entity_type$, $tail_entity_type$: the head and tail entity types (e.g., person, city)
- $true_relation$: the ground-truth relation label verbalization as defined in (Han et al., 2022a)
- $definition$: a manually written relation definition derived primarily from (Ellis et al., 2015; Stoica et al., 2021)
- $hypothesis$: the relation definition rewritten as a True/False statement, with $head_entity$ and $tail_entity$ inserted to ground it in the sample sentence
- $confirmation_boolean$: a Boolean indicating whether a sample's corresponding $hypothesis$ is correct
- $valid\ relation\ and\ definitions$: enumerated list of all valid relation classes ($\mathcal{R}_{valid}$) and their corresponding definitions based on $head_entity_type$ and $tail_entity_type$
- $relation\ verbalization\ list$: enumerated list of all valid relation classes ($\mathcal{R}_{valid}$) based on $head_entity_type$ and $tail_entity_type$
- $cot_classification$: LLM-generated CoT explanation for relation classification
- $cot_confirmation$: LLM-generated CoT explanation for True/False relation confirmation labels

```
Using the relationship definition provided, explain step-by-step why the relationship between the
head entity '{head_entity}' and the tail entity '{tail_entity}' is classified as '{true_relation}'.
The definition states that '{true_relation}' refers to: {definition}.

Please base your explanation solely on the evidence provided in the sentence: "{sentence}"

List each piece of evidence from the sentence that supports this classification and explain how it
relates to the relationship definition. Make sure your explanation is clear, logical, and avoids
any assumptions not supported by the sentence.
```

Figure 5: **Teacher Stage 1 Classification Prompt.** Zero-shot prompt used by the teacher model to generate CoT explanations for fine-tuning the Stage 1 classification student model.

```
sentence: '{sentence}'

statement: '{hypothesis}'

Using only the text of the sentence provided above as evidence, provide a step by step explanation
of why the given statement is {confirmation_boolean}. We know it is {confirmation_boolean} with
absolute certainty.
```

Figure 6: **Teacher Stage 2 Confirmation Prompt.** Zero-shot prompt used by the teacher model to generate CoT explanations for fine-tuning the Stage 2 confirmation student model.

```
{valid relation and definitions}
Instructions: Based strictly on the context within the sentence and the definitions of each
relationship type, determine if there is a direct and explicit relationship between the "head
entity" {head_entity} and the "tail entity" {tail_entity}. Choose a relation type only if clear
evidence in the sentence aligns with one of the specified definitions. If no evidence directly
supports a defined relationship, select "no relation."
Sentence: {sentence}
Head entity: {head_entity}
Head entity type: {head_entity_type}
Tail entity: {tail_entity}
Tail entity type: {tail_entity_type}
Relation types: {relation verbalization list}
Question: From the list of possible relation types, which is the relation type between the "head
entity" {head_entity} and the "tail entity" {tail_entity}? Ensure that the choice matches the
definitions and has explicit evidence from the sentence.
Format your prediction as follows: [REASONING] chain of thought reasoning here [RELATION] predicted
relation type here.
### relation type prediction: [REASONING] {cot_classification} [RELATION] {true_relation}
```

Figure 7: **Student Stage 1 Classification Prompt.** Student prompt to generate Stage 1 relation classification. During inference, the prompt ends after '### relation type prediction: '.

```
sentence: '{sentence}'
statement: '{hypothesis}'

You are given a sentence and a statement.
Your task is to decide whether the statement can be reasonably inferred from the sentence.

Answer True if the sentence clearly supports or entails the statement.

Answer False if the sentence does not support the statement, contradicts it, or leaves it uncertain.
Begin your answer with True or False, followed by a step by step explanation based only on the
sentence.
{confirmation_boolean}. {cot_confirmation}
```

Figure 8: **Student Stage 2 Confirmation Prompt.** Student prompt to generate Stage 2 relation confirmation. During inference, the prompt ends after "explanation based only on the sentence."

"members": Organizations or Geopolitical entities that are members of the head entity organization.
"siblings": The brothers and sisters of the head entity person. Full siblings, step-siblings and half-siblings are acceptable answers. Brothers- or sisters-in-law are not acceptable responses (they are for other family).
"spouse": The spouse(s) of the head entity person. Former spouses are acceptable answers.
"country of branch": The country of an organization's branch or office. The tail entity must be the name of a country.
"country of death": The country in which the head entity person died. The tail entity must be the name of a country.
"parents": The parents of the head entity person. In addition to biological parents, stepparents and adoptive parents are also acceptable answers.
"state or provinces of residence": Geopolitical entities at the state or province level in which the head entity person has lived, must be filled with state or province names. Residence must be lexically supported in source documents.
"top members or employees": The persons in high-level, leading positions at the head entity organization. Top Member/Employee positions should imply a level of decision-making authority over the entire head entity organization.
"dissolved": The date on which the head entity organization was dissolved. When companies merge to form a new entity, the original companies should be considered as dissolved.
"number of employees or members": The total number of people who are employed by or have membership in an organization.
"state or province of death": The geopolitical entity at state or province level in which the head entity person died. The tail entity must be a state or province name.
"origin": The nationality and/or ethnicity of the head entity person.
"children": The children of the head entity person, including adopted and step-children.
"political or religious affiliation": Ideological groups with which the organization is associated.
"city of birth": The geopolitical entity at the municipality level (city, town, or village) in which the head entity person was born. Mentions of 'native' or related synonyms indicate birth location. The tail entity must be the name of a city, town, or village.
"title": Official or unofficial name(s) of the employment or membership positions that have been held by the head entity person.
"shareholders": The head entity organization, person, or geopolitical entity has documented evidence of stock holdings or stock transactions involving the tail entity.
"employee of": The organizations or geopolitical entities (governments) by which the head entity person has been employed.
"member of": Organizations or geopolitical entities of which the head entity organization is a member itself.
"founded by": The person, organization, or geopolitical entity that founded the head entity organization.
"countries of residence": Countries in which the head entity person has lived.
"other family": Entities share an explicit familial relationship, limited to brother-in-law, sister-in-law, cousin, uncle, aunt, grandchild, grandparent.
"religion": The religion to which the head entity person has belonged. Former religions are acceptable answers.
"identity": Both entities refer to the same person, including cases where a pronoun is used.
"date of birth": The date on which the head entity person was born.
"city of branch": The geopolitical entity at the level of city, town, village in which a branch or office is located.
"alternate names": Any name used to refer to the head entity organization that is distinct from the "official" name. Alternate names may include former names, aliases, alternate spellings, acronyms, abbreviations, translations or transliterations of names, and any official designators such as stock ticker code or airline call sign.
"website": An official top level URL for the organization's website.
"cause of death": The explicit cause of death for the head entity person.
"state or province of branch": The geopolitical entity at state or province level where a branch of an organization is located. The tail entity must be a state or province name.
"schools attended": Any school (college, high school, university, etc.) that the head entity person has attended.
"country of birth": The country in which the head entity person was born. Mentions of 'native' or related synonyms indicate birth location. The tail entity must be the name of a country.
"date of death": The date of the head entity person's death.
"city of death": The geopolitical entity at the level of city, town, village in which the head entity person died. The tail entity must be a city, town, or village name.
"founded": The date on which the head entity organization was founded.
"cities of residence": Geopolitical entities at the level of city, town, or village in which the head entity person has lived. The tail entity must be the name of a city, town, or village.
"age": A reported age of the head entity person at a moment in time.
"charges": The charges or crimes (alleged or convicted) of the head entity person.
"state or province of birth": The geopolitical entity at state or province level in which the head entity person was born. Mentions of 'native' or related synonyms indicate birth location. The tail entity must be the name of a state or province.
"no relation": No clear evidence of any of the other listed relationships between the entities.

Figure 9: **List of All Relation Class Definitions.** Definitions used in classification CoT generation and Stage 1 classification model prompting.

"members": The geopolitical entity or organization tail_entity is a member of the organization head_entity.
"siblings": The individuals head_entity and tail_entity are siblings (this includes full, step-, and half-siblings, but not in-laws).
"spouse": head_entity is the spouse of tail_entity. Former spouses are valid.
"country of branch": The organization head_entity has a branch or office in the country tail_entity. tail_entity must be the name of a country.
"country of death": The person head_entity died in the country tail_entity. tail_entity must be the name of a country.
"parents": The individual tail_entity is a parent of head_entity (this includes biological, step-, or adoptive parents).
"state or provinces of residence": The person head_entity has lived at the state or province tail_entity. tail_entity is at the state or province level.
"top members or employees": The persons tail_entity is in a high-level, leading position at the organization head_entity. tail_entity has a level of decision-making authority over the organization head_entity.
"dissolved": The organization head_entity was dissolved on the date tail_entity. When companies merge to form a new entity, the original companies should be considered as dissolved.
"number of employees or members": head_entity organization has tail_entity number of employees or members.
"state or province of death": head_entity died in the state or province tail_entity. tail_entity is at the state or province level.
"origin": head_entity has nationality or ethnicity of tail_entity.
"children": tail_entity is a child (biological, adopted, or step-child) of head_entity.
"political or religious affiliation": head_entity is affiliated with the political or religious group tail_entity.
"city of birth": head_entity was born in the city of tail_entity. tail_entity is at the city level.
"title": head_entity has held the title or position of tail_entity.
"shareholders": head_entity has documented evidence of stock holdings or stock transactions involving tail_entity.
"employee of": head_entity was employed by the organization or geopolitical entity tail_entity.
"member of": Organization head_entity is a member of the organization or geopolitical entity tail_entity.
"founded by": head_entity was founded by tail_entity.
"countries of residence": head_entity has lived in the country tail_entity. tail_entity must be the name of a country.
"other family": head_entity is the brother-in-law, sister-in-law, cousin, uncle, aunt, grandparent, grandson, or granddaughter of tail_entity.
"religion": head_entity belonged to the religion tail_entity.
"identity": head_entity and tail_entity refer to the same person.
"date of birth": head_entity was born on the date tail_entity.
"city of branch": head_entity has a branch or office located in the city tail_entity. tail_entity is at the city level.
"alternate names": tail_entity is a name used to refer to the head_entity that is distinct from the "official" name. This may include former names, aliases, alternate spellings, acronyms, abbreviations, translations or transliterations of names, and any official designators such as stock ticker code or airline call sign.
"website": tail_entity is the official website of the organization head_entity.
"cause of death": head_entity died of the cause of death tail_entity.
"state or province of branch": head_entity has a branch or office located in the state or province tail_entity. tail_entity is at the state or province level.
"schools attended": head_entity attended the school tail_entity.
"country of birth": head_entity was born in the country tail_entity. tail_entity must be the name of a country.
"date of death": head_entity died on the date tail_entity.
"city of death": head_entity died in the city of tail_entity. tail_entity is at the city level.
"founded": The organization head_entity was founded on the date tail_entity.
"cities of residence": head_entity lived in the city tail_entity.
"age": head_entity is the age of tail_entity.
"charges": head_entity was charged with or convicted of the crime tail_entity.
"state or province of birth": head_entity was born in the state or province tail_entity. tail_entity is at the state or province level.

Figure 10: **List of All Relation Class True/False Statements.** Statements used in confirmation CoT generation and Stage 2 confirmation model prompting. When used in prompts, $head_entity$ and $tail_entity$ are replaced with their sample values.

C Relation Classification Performance

Figure 11 shows macro and weighted performance metrics across all experiments, aggregated by backbone, k, and dataset configuration. Overlapping Stage 1 and Stage 2 bars highlight the delta in performance between stages. The bar for the stage with the smaller value is plotted on top of the bar with the larger value for visibility.

In the first row, there are very minor performance gaps between Stage 1 and Stage 2 weighted-precision results, reflecting the accuracy of the Stage 2 confirmation model, with only a small set of named-relation samples being incorrectly reassigned to "no relation." Notably, the two-stage design maintains strong weighted-precision, while substantially improving weighted-recall, showing the model accurately reassigns the over-predicted "no relation" samples. This combination of a maintenance in strong precision and improved recall results in a large increase in weighted-F1 across all experimental configurations.

The second row shows macro-averaged performance metrics. Stage 2 reduces false positives present across classes on average, consistently improving macro-precision. However, macro-recall drops across all configurations as the confirmation stage reassigns some correctly predicted named-relations from Stage 1 to the "no relation" class. Despite this, the improvement in precision outweighs the reduction in recall as macro-F1 improved across all experimental settings.

We illustrate the Stage 1 to Stage 2 macro and weighted metric trade-offs in Figure 13, aggregated over student-LLM architectures to highlight the consistency of the trend across k values and datasets (Section 6.2).

Figure 12 shows consistent improvements in micro-F1 from Stage 1 to Stage 2 across all experimental settings. This gain is driven by the accurate reassignment of over-predicted samples to the correct "no relation" class via the Stage 2 confirmation model, as supported by the maintenance of weighted precision between stages, while correctly predicted named-relation samples remain largely preserved, reflected by the minor decrease in macro-recall between stages. The magnitude of the improvement between stages is largely driven by the prevalence of "no relation" samples in each dataset and the accuracy of the Stage 2 confirmation model.

D Effect of Stage 2 Confirmation Model Across Few-Shot Settings

			Stage 1			Stage 2		
Avg	k	Dataset	P	R	F1	P	R	F1
Macro	8	TACRED	27.1	77.1	35.3	49.6	59.8	50.7
		TACREV	25.7	79.6	33.7	50.7	66.3	52.9
		Re-TACRED	41.5	80.8	49.9	67.3	65.9	63.2
Weighted	8	TACRED	84.5	34.2	35.3	84.2	78.6	80.3
		TACREV	86.0	32.5	33.4	87.1	79.5	81.8
		Re-TACRED	82.4	56.3	57.9	82.3	80.1	80.1
Micro	8	TACRED		34.2			78.6	
		TACREV		32.5			79.5	
		Re-TACRED		56.3			80.1	
Macro	16	TACRED	26.9	78.7	35.6	50.3	63.0	52.2
		TACREV	26.4	81.7	34.9	51.5	69.6	55.3
		Re-TACRED	42.9	83.1	51.7	66.6	69.7	65.0
Weighted	16	TACRED	84.6	33.6	34.4	84.9	79.1	80.9
		TACREV	86.6	34.6	36.7	87.8	82.1	83.9
		Re-TACRED	83.5	59.9	62.4	83.0	80.7	80.9
Micro	16	TACRED		33.6			79.1	
		TACREV		34.6			82.1	
		Re-TACRED		59.9			80.7	
Macro	32	TACRED	27.3	79.8	36.4	49.7	65.1	53.1
		TACREV	27.5	83.2	36.7	51.0	72.2	55.8
		Re-TACRED	44.3	85.6	54.1	67.1	73.2	67.0
Weighted	32	TACRED	85.1	36.3	38.4	85.4	79.0	80.9
		TACREV	87.2	38.5	41.5	88.4	81.0	83.3
		Re-TACRED	84.3	63.2	65.4	84.1	81.5	81.8
Micro	32	TACRED		36.3			79.0	
		TACREV		38.5			81.0	
		Re-TACRED		63.2			81.5	

Table 6: Stage 1 and Stage 2 performance metrics at $k \in \{8, 16, 32\}$ averaged across seeds and backbone architectures.

As shown in Figure 13 and Table 6, Stage 1 achieves strong macro-averaged recall across each dataset and k setting. This indicates that the retrieval stage successfully identifies named-relation instances across data regimes. However, Stage 1 weighted-recall, dominated by the majority "no relation" class, substantially underperforms macro-averaged recall across all k settings and datasets. This reveals that Stage 1 over-predicts named relations at the expense of "no relation" class across each data setting. Additional data availability results in Stage 1 improvement gains, with mean macro-recall improvements from 79.2 at $k = 8$ to 81.2 at $k = 16$ and 82.9 at $k = 32$. Micro-recall improves similarly from 41.0 at $k = 8$ to 42.7 at $k = 16$ and 46.0 at $k = 32$, showing the additional data helps improve named-relation recall while also

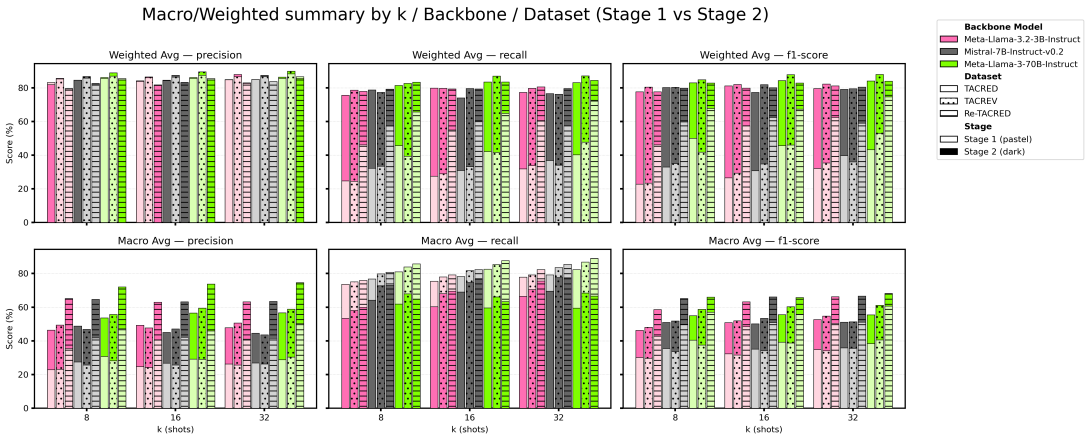

Figure 11: **Macro and weighted precision, recall and F1 across backbone models, k settings, and datasets.** Stage 1 and Stage 2 results are shown as overlapping bars to highlight the performance delta between stages. Datasets are distinguished by bar pattern, backbone models by color and stage by color saturation (Stage 1, pastel colors and Stage 2, saturated colors). Results reflect performance averaged over 5 random seeds. Bars smaller in value are plotted on top of bars larger in value.

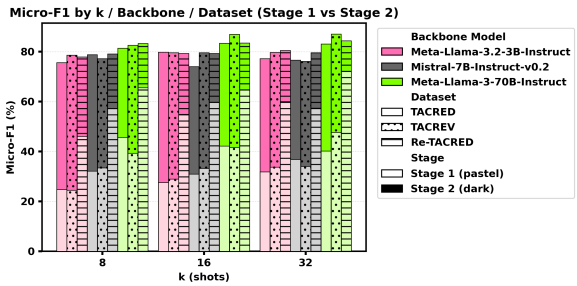

Figure 12: **Micro-F1 across backbone models, k settings, and datasets.**

helping mitigate the over-prediction problem but does not fully resolve it.

Stage 2 effectively addresses the over-prediction problem across the data regimes tested, evidenced by average weighted recall increases of $+38.4$, $+37.9$, and $+34.5$ for $k = 8, 16, 32$ respectively. At the macro level, the precision-recall trade-off yields net macro-F1 gains of $+16.0$, $+16.8$, and $+16.2$ at $k = 8, 16, 32$. Additional data improves macro-F1 primarily in Stage 1, with Stage 2 performance more similar across k values.

The impact on micro-F1 is most striking: improving from 41.0 to 79.4 at k=8, 42.7 to 80.6 at k=16, and 46.0 to 80.5 at k=32 when averaged across datasets and backbones. Taken together, these results validate the two-stage design: named-relation retrieval followed by targeted True/False confirmation effectively balances recall across relation classes with accurate "no relation" rejection.

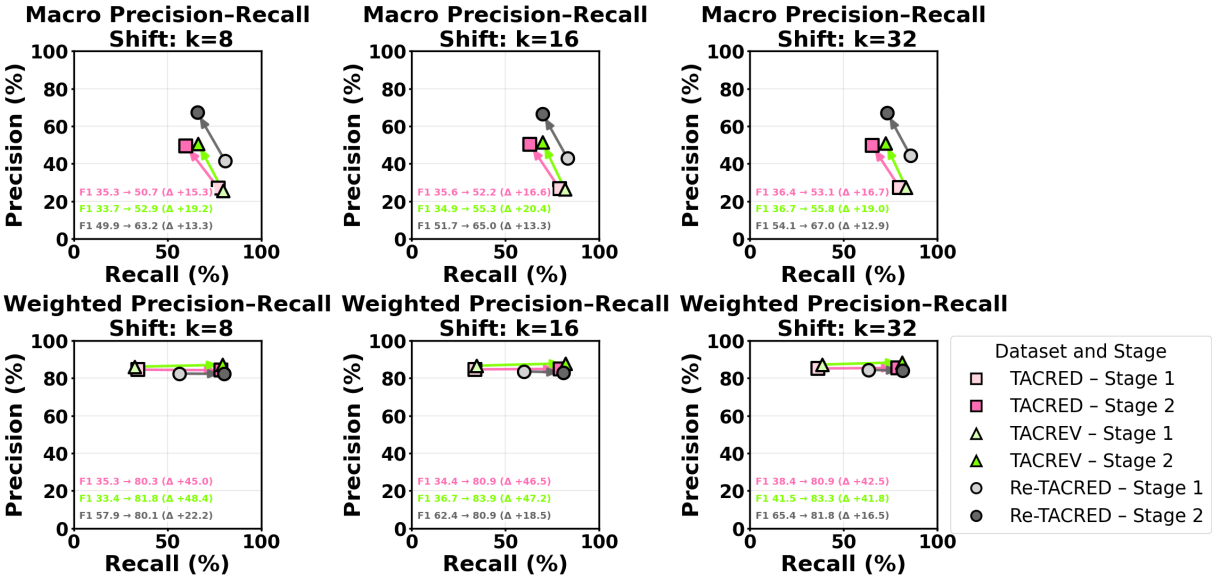

Figure 13: Precision-Recall shift from Stage 1 (Classification) to Stage 2 (Confirmation) of Distill-RE at k=8, 16, 32, averaged across student LLM architectures. Top row: macro-averaged; Bottom row: weighted-averaged. Arrows indicate the direction and magnitude of the shift, with Δ F1 annotated per dataset.