From Cross-Task Examples to In-Task Prompts: A Graph-Based Pseudo-Labeling Framework for In-context Learning

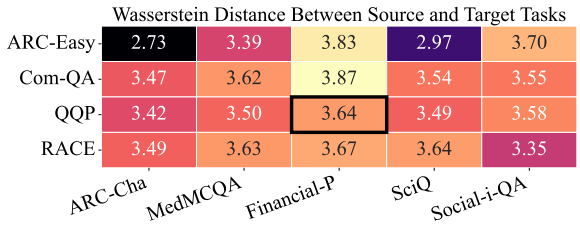
Zihan Chen, Song Wang, Xingbo Fu, Chengshuai Shi, Zhenyu Lei, Cong Shen, Jundong Li
Department of ECE, University of Virginia, Charlottesville, VA, USA
{brf3rx,sw3wv,xf3av,cs7ync,vjd5zr,cs7dt,jl6qk}@virginia.edu

Abstract

The capability of in-context learning (ICL) enables large language models (LLMs) to perform novel tasks without parameter updates by conditioning on a few input-output examples. However, collecting high-quality examples for new or challenging tasks can be costly and labor-intensive. In this work, we propose a cost-efficient two-stage pipeline that reduces reliance on LLMs for data labeling. Our approach first leverages readily available cross-task examples to prompt an LLM and pseudo-label a small set of target task instances. We then introduce a graph-based label propagation method that spreads label information to the remaining target examples without additional LLM queries. The resulting fully pseudo-labeled dataset is used to construct in-task demonstrations for ICL. This pipeline combines the flexibility of cross-task supervision with the scalability of LLM-free propagation. Experiments across five tasks demonstrate that our method achieves strong performance while lowering labeling costs. Our code is available at https://github.com/Chen-1031/Cross-Task-ICL.

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities across a wide range of natural language processing tasks (Zhao et al., 2023; Chang et al., 2024), including semantic parsing (Li et al., 2021; Wolfson et al., 2020) and commonsense reasoning (Talmor et al., 2019; Zellers et al., 2019; Lei et al., 2025b,a). However, the substantial computational cost of retraining or fine-tuning these models limits their practicality for novel tasks (Hu et al., 2021; Liu et al., 2022; Zaken et al., 2022). Fortunately, LLMs possess an emergent ability known as In-Context Learning (ICL) (Wang et al., 2024c,a; Chen et al.), wherein the model can perform new tasks by conditioning on a few input-output pairs (i.e., demonstrations) during inference, without updating model parameters (Brown et al., 2020).



Task	Label Space
QQP	{duplicate, not duplicate}
Financial-Phrasebank	{positive, neutral, negative}

Figure 1: (Top) Wasserstein distance between source (column) and target (row) task example embeddings. (Bottom) Examples of task label spaces.

Despite its promise, the effectiveness of ICL heavily relies on high-quality labeled examples for the target task. For novel or data-scarce tasks, recent work has explored using LLMs with zero-shot prompts (Zhang et al., 2025; Wan et al., 2024; Shi et al., 2024) or relying on human annotators (Mikulová et al., 2023; Wang et al., 2024b) to obtain pseudo-labeled examples, which are then used as demonstrations for ICL. Yet both approaches have drawbacks: LLMs can be unreliable on unfamiliar tasks, while human annotation introduces additional time and labor costs (Su et al., 2022a). To overcome this, recent efforts have turned to leveraging well-established, high-resource source tasks to construct demonstrations (Tanwar et al., 2023; Raffel et al., 2020). When the examples used for ICL are drawn from a different task than the target, the setting is referred to as cross-task ICL. For example, Chatterjee et al. (2024) select examples from source tasks based on embedding similarity and demonstrate that such cross-task examples can significantly improve ICL performance, highlighting cross-task ICL as a promising approach for pseudo-labeling.

However, as illustrated in Figure 1, the data dis-

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 3395–3410
November 4-9, 2025 ©2025 Association for Computational Linguistics

tributions of source and target tasks often differ significantly, and the label spaces can be misaligned, even between tasks with similar distributions. This raises a key limitation: selecting cross-task examples based solely on embedding similarity is insufficient for reliable pseudo-labeling of target samples.

To address this, we draw inspiration from the graph mining literature, which shows that structural properties of graphs can generalize across domains even when feature spaces are heterogeneous (Qiu et al., 2020; Leskovec et al., 2005; Hamilton et al., 2017). Based on this, we propose GraphSim, a graph-based example selection method that augments text embeddings with structural information captured through graph aggregation—a process where each node's representation is updated by aggregating information from its neighbors. These structure-aware embeddings yield more robust similarity metrics across tasks, allowing for better example selection. To further address the label space mismatch across tasks, we propose GLIP (Graph-based Label Information Propagation), a label information propagation framework that uses a small set of pseudo-labeled target examples (obtained via GraphSim and an LLM) to infer labels for the remaining unlabeled target examples. The resulting pseudo-labeled target set can then be used as high-quality demonstrations for in-context learning. Notably, our pipeline is cost-efficient and adaptable: it only requires a small number of LLM calls for pseudo-labeling and leverages lightweight graph-based propagation for the rest, making it practical for real-world deployment on novel tasks. Our contributions can be summarized as follows:

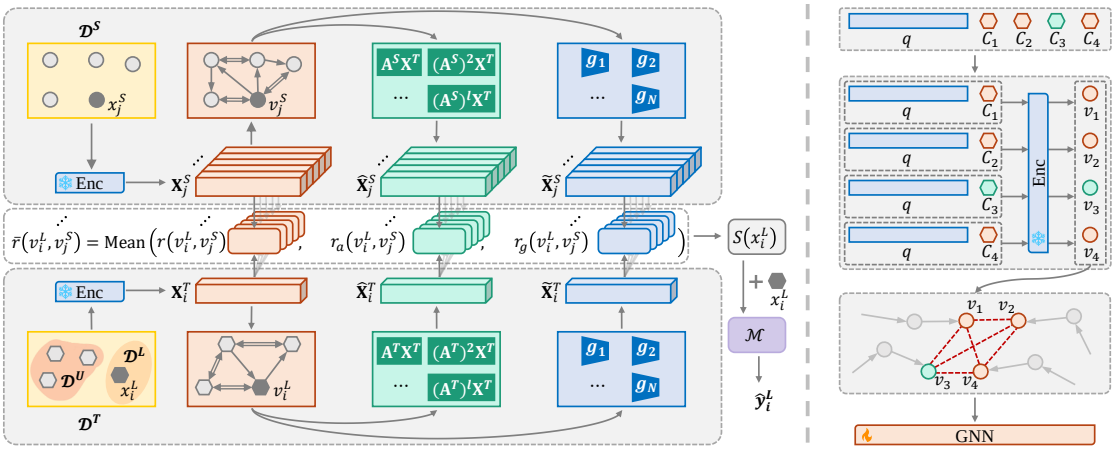
- **Problem Formulation**: We propose a novel problem formulation that utilizes examples from high-resource source tasks to pseudo-label examples from a novel target task. This enables in-context learning without requiring extensive manual annotation or large-scale LLM inference on the target task.
- **Methodological Innovation**: We introduce a two-stage graph-based pipeline that addresses both cross-task example selection and label propagation. First, we propose GraphSim, a structure-aware similarity metric for selecting source examples that are more transferable across tasks. Second, we design GLIP, which efficiently propagates labels from a few LLM-labeled target examples to the rest of the target dataset, mitigating label space misalignment.
- **Empirical Validation**: Comprehensive experiments on five target tasks with five different-sized LLMs demonstrate that our method outperforms existing cross-task baselines and approaches the performance of in-task upper bounds with light reliance on LLMs, highlighting both its effectiveness and efficiency.

2 Related Works

In-Context Learning. In-context learning (ICL) (Brown et al., 2020) equips large language models (LLMs) with the ability to leverage a handful of input-output demonstrations for reasoning. ICL has proven remarkably successful in handling complex tasks, including summarization (Jain et al., 2023; Baek et al., 2024) and question answering (Lee et al., 2024). To improve ICL performance, many works have explored adaptive strategies for selecting effective demonstrations (Lu et al., 2021; Zhao et al., 2021; Ye et al., 2023; Chen et al., 2024), such as retrieving semantically similar examples (Liu et al., 2021). However, these methods typically assume that the demonstrations and the input come from the same task (i.e., in-task ICL) and rely on access to a large pool of labeled examples. This assumption limits their applicability in low-resource or novel task settings, where collecting high-quality annotations is costly and impractical.

Cross-task In-Context Learning. Due to the large disparity in annotation availability between novel tasks and well-established ones, leveraging high-resource source tasks to improve performance on low-resource target tasks has become an appealing direction. Prior work has explored cross-task ICL in various settings, such as cross-lingual tasks (Tanwar et al., 2023), multi-task learning (Zhang et al., 2022; Raffel et al., 2020), and prompt generation for downstream tasks (Zou et al., 2023). More recently, Chatterjee et al. (2024) demonstrated that LLMs can benefit significantly from cross-task ICL prompts and showed the potential of generating pseudo-labels for in-task examples. However, their method relies on semantic similarity for example selection and labels only a small subset of target examples, leaving most unlabeled data unused. In contrast, our cost-efficient pipeline combines LLM-based pseudo-labeling of a small seed set with graph-based label information propagation, enabling scalable construction of high-quality in-task demonstrations.

.



(a) Graph-based Example Selection for Cross-task Labeling (GraphSim) (b) Label Information Propagation

Figure 2: Overview of our proposed pipeline for cross-task pseudo-labeling. We first use (a) **GraphSim** to select relevant examples from the source task to pseudo-label a small set of target task examples $\mathcal{D}^L$ via ICL. Then, we apply (b) **GLIP**, a graph-based label propagation method, to infer labels for the remaining unlabeled target samples $\mathcal{D}^U$. The resulting fully pseudo-labeled target set is used to construct in-task examples for in-task ICL

3 Preliminaries

3.1 Graph Neural Networks

Let $\mathcal{G} = (\mathcal{V}, \mathbf{A}, \mathbf{X})$ denote an attributed graph with a set of nodes $\mathcal{V} = \{v_1, v_2, \cdots, v_{|\mathcal{V}|}\}$. $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the feature matrix where each row $\mathbf{X}_i \in \mathbb{R}^d$ is the d-dimensional feature vector of node $v_i \in \mathcal{V}$. $\mathbf{A} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the adjacency matrix, where each entry $\mathbf{A}_{ij} = 1$ if nodes v_i and v_j are connected by an edge; otherwise, $\mathbf{A}_{ij} = 0$. GNNs typically follow a message-passing framework (Kipf and Welling, 2017; Veličković et al., 2018), where each node iteratively aggregates information from its neighbors. At the l-th layer, node v_i's representation $\mathbf{H}_i^{(l)}$ is updated as:

$$\mathbf{H}_i^{(l)} = g^{(l)}(\mathbf{H}_i^{(l-1)}, \{\mathbf{H}_j^{(l-1)} : v_j \in \mathcal{N}(v_i)\}; \theta^{(l)}), \quad (1)$$

where $\mathcal{N}(v_i)$ denotes the neighbors of v_i, and $g^{(l)}$ is the aggregation function with parameters $\theta^{(l)}$. We initialize $\mathbf{H}_i^{(0)}$ as the node feature, i.e., $\mathbf{H}_i^{(0)} = \mathbf{X}_i$.

3.2 In-Contex Learning

In-Context Learning (ICL) enables LLMs to perform a new task simply by conditioning on a few input-output examples without any fine-tuning (Brown et al., 2020). Given a prompt composed of k examples $\{(x_j, y_j)\}_{j=1}^k$ and a new query input x_{query}, the model generates a prediction $\hat{y}_{\text{query}}$ by autoregressively decoding the next token(s):

$$\hat{y}_{\text{query}} = \mathcal{M}((x_i, y_1), ..., (x_k, y_k), x_{\text{query}}),$$

where $\mathcal{M}$ is the frozen LLM. The effectiveness of ICL largely depends on the quality and relevance of the selected examples (Liu et al., 2021).

4 Methodology

4.1 Problem Setup

We formulate each input-output pair (x, y) as a multiple-choice question, where $x = (q, \{C_1, \ldots, C_n\})$ consists of a query and n candidate choices, and y is the correct answer.

Let $\mathcal{D}^S = \{(x_i^S, y_i^S)\}_{i=1}^{|\mathcal{D}^S|}$ be a labeled source dataset, and let $\mathcal{D}^T = \{(x_i^T, y_i^T)\}_{i=1}^{|\mathcal{D}^T|} = \mathcal{D}^L \cup \mathcal{D}^U$ be the target dataset, with $\mathcal{D}^L$ denoting a small subset used for cross-task in-context labeling, and $\mathcal{D}^U$ the remaining unlabeled examples. For each $x^L \in \mathcal{D}^L$, we use a pretrained language model $\mathcal{M}$ to generate a pseudo-label $\hat{y}^L$ conditioned on selected source examples $\mathcal{S}(x^L) \subseteq \mathcal{D}^S$:

$$\hat{y}^L = \mathcal{M}(\mathcal{S}(x^L), x^L). \quad (2)$$

To avoid labeling the full target set with LLMs, we propagate label information from $\mathcal{D}^L$ to $\mathcal{D}^U$ using an LLM-free algorithm. The resulting pseudo-labeled target set is then used to construct in-task examples for ICL.

4.2 Graph-based Example Selection for Cross-task Labeling (GraphSim)

In this section, we propose a graph-based method, GraphSim, to select cross-task examples for pseudo-labeling. Traditional ICL selects examples

based on embedding similarity, assuming query and examples come from the same task. This assumption fails in cross-task settings, where data distributions and label spaces often differ (Figure 1).

To address this, we leverage structural patterns in data via graph-based modeling. Our approach GraphSim is motivated by findings in graph mining, which suggest that structural properties of graphs can generalize across domains, even when feature spaces are heterogeneous (Qiu et al., 2020; Leskovec et al., 2005). Specifically, we independently construct task-specific graphs for both source and target tasks. We then apply graph-based aggregation to enrich the embeddings with structural information. These structure-aware embeddings better capture task-level semantics, enabling more meaningful similarity computations across tasks. To further enhance representativeness, GraphSim incorporates a set of randomly initialized GNNs to introduce diverse views during aggregation, enhancing the cross-task example selection.

Specifically, given the source task dataset $\mathcal{D}^S$, we construct a graph $\mathcal{G}^S = (\mathcal{V}^S, \mathbf{A}^S, \mathbf{X}^S)$ to model the relationships among samples, where each node $v_i^S \in \mathcal{V}^S$ corresponds to a sample $x_i^S \in \mathcal{D}^S$ whose feature vector is obtained through a pre-trained text encoder, i.e., $\mathbf{X}_i^S = \texttt{Enc}(x_i^S)$. To obtain $\mathbf{A}^S$, we connect each node to its top-k most relevant neighbors based on a pairwise relevance score. Specifically, the relevance between two samples is computed as the cosine similarity of their encoded representations:

$$r(v_i^S, v_j^S) = \cos(\mathbf{X}_i^S, \mathbf{X}_j^S). \tag{3}$$

Afterwards, each entry $\mathbf{A}_{ij}^S \in \mathbf{A}^S$ of the source graph is computed as:

$$\mathbf{A}_{ij}^S = \begin{cases} 1, & \text{if } r(v_i^S, v_j^S) \in \text{Top-}k\left\{r(v_i^S, v_j^S)\right\}, \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

ensuring that only the k most relevant connections are retained for each node. Similarly, we can construct the graph $\mathcal{G}^T = (\mathcal{V}^T, \mathbf{A}^T, \mathbf{X}^T)$ for the target task using the target dataset $\mathcal{D}^T$.

To incorporate structural information into the node representations, we apply two types of aggregation: (1) Adjacency-based aggregation, and (2) GNN-based aggregation.

The adjacency-based aggregation captures multi-hop neighborhood information by applying powers of $\mathbf{A}^S$ to $\mathbf{X}^S$ and concatenating the results:

$$\hat{\mathbf{X}}^S = [\mathbf{A}^S\mathbf{X}^S || \left(\mathbf{A}^S\right)^2 \cdot \mathbf{X}^S || \cdots || \left(\mathbf{A}^S\right)^l \mathbf{X}^S],$$

where l controls the number of hops (i.e., the neighborhood depth). This allows each node to gather information from its l-hop neighbors and encode structural patterns beyond immediate connections.

Beyond adjacency-based aggregation, we further capture structural information using a set of randomly initialized GNNs. The key intuition is that if two nodes exhibit similar structural patterns, their aggregated representations from a GNN will also be similar. To achieve this without requiring training, we initialize a collection of GNNs $\{g_1, g_2, \ldots, g_N\}$, each with a different number of layers and independently randomized parameters. This design offers two main advantages: (i) GNNs with varying depths and parameterizations capture structural features from diverse perspectives; (ii) Random initialization avoids training overhead, making our method efficient, easily scalable, and broadly applicable across tasks. The GNN-based aggregation is then formulated as:

$$\tilde{\mathbf{X}}^S = [g_1(\mathbf{A}^S, \mathbf{X}^S) || \cdots || g_N(\mathbf{A}^S, \mathbf{X}^S)].$$

With the proposed two aggregation methods, we generate augmented representations for the source and target tasks separately as follows:

$$\mathcal{X}^S = \{(\mathbf{X}^S, \hat{\mathbf{X}}^S, \tilde{\mathbf{X}}^S)\},$$
$$\mathcal{X}^T = \{(\mathbf{X}^T, \hat{\mathbf{X}}^T, \tilde{\mathbf{X}}^T)\},$$

where each tuple combines the original, adjacency-based, and GNN-based representations of a sample.

We then compute the cross-task similarity between nodes $v_i^T \in \mathcal{V}^T$ and $v_j^S \in \mathcal{V}^S$ by averaging the similarities of each type of embedding:

$$\begin{aligned} \bar{r}(v_i^T, v_j^S) = &\texttt{Mean}(r(v_i^T, v_j^S), \\ &r_a(v_i^T, v_j^S), r_g(v_i^T, v_j^S)), \end{aligned}$$

where $\texttt{Mean}(\cdot,\cdot,\cdot)$ denotes the average operation. r_a and r_g are similarity scores based on adjacency- and GNN-augmented embeddings, respectively. Averaging the separately computed similarities offers computational efficiency and helps mitigate the curse of dimensionality that may arise from direct concatenation.

Using the cross-task similarity $\bar{r}$, we select the source examples $\mathcal{S}(x_i^L) \subseteq \mathcal{D}^S$ for $x_i^L \in \mathcal{D}^L$ as:

$$\mathcal{S}(x_i^L) = \{x_j^S | \bar{r}(x_j^S, x_i^L) \in \text{Top-}K(\{\bar{r}(x^S, x_i^L)\})\}.$$

Following Equation 2, we use the LLM's prediction $\hat{y}_i^L$ to construct a pseudo-labeled set $\bar{\mathcal{D}}^L = \{(x_i^L, \hat{y}_i^L)\}$, which provides label information for subsequent in-task information propagation.

4.3 Label Information Propagation (GLIP)

Prior work shows that increasing in-context examples generally boosts ICL performance by providing richer task-specific signals (Agarwal et al., 2024). While labeling the entire target set $\mathcal{D}^T$ via cross-task ICL is possible, it incurs high cost as the number of LLM calls grows linearly with data size. To reduce this cost, we adopt graph-based semi-supervised learning, where labels are propagated from a small labeled set. However, applying this to QA tasks is nontrivial—labels in multiple-choice QA must be invariant to choice order.

To address these challenges while leveraging the strengths of graph-based learning, we propose GLIP, a tailored graph construction method with two types of edges and a GNN trained to perform label information propagation.

We begin with graph construction. To ensure invariance to choice order, we encode QA examples in a query-choice pairwise way. Specifically, for each pseudo-labeled example $(x, \hat{y}) = (q, \{C_1, C_2, ..., C_n\}, \hat{y}) \in \bar{\mathcal{D}}^L$, we generate n nodes with features $\{\texttt{Enc}([q, C_1]), ..., \texttt{Enc}([q, C_n])\}$. The multiple-choice QA task is thus transformed into a multi-node binary classification problem, where each node receives a label of 1 (correct) or 0 (incorrect). For example, if $\hat{y} = C_3$, then we label the node with feature $\texttt{Enc}([q, C_3])$ as 1, and the remaining nodes are labeled as 0. For unlabeled examples in $\mathcal{D}^U$, we construct nodes in the same way but leave their labels unassigned.

We then define two types of edges to capture task-specific structure: (i) Similarity-based positive edges $\mathcal{E}_{pos}$: Constructed using the relevance scores between query-choice pairs, as described in Section 4.2. These edges *only* connect nodes from different queries to model semantic relationships across both labeled and unlabeled samples, which is critical for pseudo-labeling unlabeled samples. (ii) Mutual exclusion negative edges $\mathcal{E}_{neg}$: These encode Mutual Exclusion Constraints (MEC) (Su et al., 2022b) by connecting nodes within the same query. Since only one choice can be correct, these edges penalize configurations where multiple nodes for the same question are labeled as correct (i.e., 1) during training, enforcing consistency in the multi-choice setting.

Following the construction of nodes and edges, we obtain a graph $\mathcal{G}$ composed of labeled and unlabeled nodes. To perform semi-supervised learning, we train a graph neural network (GNN) $\tilde{g}$ on $\mathcal{G}$ by minimizing the objective $\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{MEC}$, where $\mathcal{L}_{CE}$ is the standard cross-entropy loss over labeled nodes, and $\mathcal{L}_{MEC}$ is a MEC loss designed to reduce the similarity between nodes connected byy $\mathcal{E}_{neg}$:

$$\mathcal{L}_{MEC} = -\frac{1}{|\mathcal{E}_{neg}|} \sum_{i,j \in \mathcal{E}_{neg}} \langle \mathbf{h}_i, \mathbf{h}_j \rangle,$$

where $\mathbf{h}_i$ and $\mathbf{h}_j$ are the output embeddings of nodes i and j from the GNN $\tilde{g}$, respectively.

Once trained, $\tilde{g}$ is used to predict labels for unlabeled nodes in $\mathcal{G}$. For each unlabeled input $x \in \mathcal{D}^U$, we determine its predicted answer by selecting the choice with the highest logit:

$$\hat{y} = \arg\max_{i \in [n]} \tilde{g}(Enc([q, C_i]).$$

After combining the pseudo-labeled sets from the cross-task labeling stage and the graph-based label propagation, we obtain a fully pseudo-labeled target dataset $\bar{\mathcal{D}}^T = \bar{\mathcal{D}}^L \cup \bar{\mathcal{D}}^U$. This augmented set can then be used in a traditional similarity-based in-context learning pipeline for final inference on the target task.

5 Experiment

5.1 Experiment setup

Datasets and experimental setup. We follow the same source and target dataset setting as in Chatterjee et al. (2024). Detailed task descriptions for both source and target tasks are provided in Appendix A. We focus on the single-source-task scenario and select the best-performing source task for each target task based on the results reported in Chatterjee et al. (2024)[1]. The source-target task pairs used in our experiment are listed in Table 1.

For consistency, we standardize dataset sizes: we sample 10,000 examples from each source task ($|\mathcal{D}^S| = 10,000$) and construct a 500-example pool for each target task ($|\mathcal{D}^T| = 500$), among

[1]Identifying the optimal source remains an open challenge. One promising direction is to select source tasks based on the similarity of final-layer hidden states between source and target task definitions (Chatterjee et al., 2024).

which 100 examples are used for cross-task labeling and the remaining 400 are used for label propagation. An additional disjoint set of 500 target examples is reserved for final inference evaluation.

We use Sentence-BERT (Reimers and Gurevych, 2019) as the text encoder $\texttt{Enc}(\cdot)$. Unless otherwise specified, we set $k = 20$ for graph construction and $l = 2$ for adjacency-based aggregation. For GNN-based aggregation, we initialize $N = 4$ randomly initialized GCNs (Kipf and Welling, 2017): two with a single layer and two with two layers, each with hidden size 128. For label information propagation, we use a two-layer GAT (Veličković et al., 2018) with hidden size 64 as the backbone model. We train it for 25 epochs using the Adam optimizer with a learning rate of 0.005. The loss balancing coefficient is set to $\lambda = 0.4$. For LLM-based cross-task labeling, we follow Chatterjee et al. (2024) and evaluate with LLaMA2-7B, LLaMA2-13B (Touvron et al., 2023), and GPT-3.5 (OpenAI, 2023). Additional results using LLaMA3-8B (AI@Meta, 2024) and GPT-4o (Hurst et al., 2024) are presented in the Appendix B.

Table 1: Source-Target task pairs used in the experiment.

LLM	Target Task	Source Task
LLaMA2-7B **LLaMA3-8B**	ARC-Challenge MedMCQA Financial-Phrasebank SciQ Social-i-QA	ARC-Easy Commonsense-QA SST2 Commonsense-QA RACE
LLaMA2-13B	ARC-Challenge MedMCQA Financial-Phrasebank SciQ Social-i-QA	ARC-Easy RACE QQP Commonsense-QA RACE
GPT-3.5 **GPT-4o**	ARC-Challenge MedMCQA Financial-Phrasebank SciQ Social-i-QA	RACE BoolQ AG-news RACE RACE

Baselines. We compare our pipeline against three types of baselines: i) **Zero-shot**: The LLM is prompted with only the task instruction. ii) **Cross-task ICL**: Includes EmbSim and GraphSim, which retrieve examples from a source task. EmbSim only uses embedding similarity, while GraphSim applies our graph-based method in Section 4.2. iii) **In-task ICL**: Includes $\mathcal{L}_{\text{LLM}}$, GLIP, and Oracle. These differ in how they construct the labeled target pool $\mathcal{D}^T$. $\mathcal{L}_{\text{LLM}}$ uses labeled $\mathcal{D}^L$ to perform in-task ICL and pseudo-label $\mathcal{D}^U$; GLIP applies our proposed method in Section 4.3) to label $\mathcal{D}^U$; and Oracle represents an idealized case where the entire $\mathcal{D}^T$ is assumed to be gold-labeled and directly used as the demonstration pool. A summary of the differences between these methods is provided in Table 2.

Table 2: Source of examples used by different baselines. Red lines indicate cross-task ICL settings, and Blue lines indicate in-task ICL settings.

Method	$\mathcal{D}^S$	$\mathcal{D}^L$	$\mathcal{D}^U$
Zero-shot	✗	✗	✗
EmbSim	Ground Truth	✗	✗
GraphSim	Ground Truth	✗	✗
$\mathcal{L}_{LLM}$	✗	Ground Truth	In-task ICL
GLIP	✗	Ground Truth	GLIP
Ours	✗	GraphSim	GLIP
Oracle	✗	Ground Truth	Ground Truth

5.2 Main Result

Table 3 presents the performance of our proposed pipeline and six baselines across five target tasks under one-shot and four-shot in-context learning settings. We highlight the following key observations: (i) **Graph-based similarity improves cross-task in-context learning.** Both EmbSim and GraphSim operate in the cross-task ICL setting. Across all tasks, GraphSim consistently outperforms EmbSim, demonstrating that augmenting input representations with structural information, shared across different domains, helps identify more relevant examples for cross-task labeling. This leads to higher-quality pseudo-labels for the target task. (ii) **Label information propagation outperforms LLM-based pseudo-labeling.** $\mathcal{L}_{LLM}$ and GLIP use the same golden-labeled set $\mathcal{D}^L$, but differ in how they generate labels for $\mathcal{D}^U$: $\mathcal{L}_{LLM}$ uses in-task ICL with the LLM, while GLIP applies our proposed graph-based propagation method. GLIP consistently outperforms $\mathcal{L}_{LLM}$ and achieves performance closer to the Oracle, suggesting that label propagation is more reliable — particularly for challenging tasks where LLM-generated labels can be noisy and unreliable. (iii) **Our full pipeline is both effective and cost-efficient.** The combination of graph-based cross-task labeling and label propagation results in high-quality pseudo-labeled target sets, enabling strong ICL performance on novel tasks. Our method significantly reduces reliance on LLM inference during labeling, making it more efficient and practical in real-world, resource-constrained settings.

Table 3: Performance comparison of our pipeline and six baselines on five target tasks under one-shot and four-shot in-context learning settings. Dashed lines separate cross-task ICL results from in-task ICL results. Bold font highlights the best performance within each category, while underlined values indicate the second-best results within the in-task ICL group for clearer analysis.

Method	ARC-Challenge		MedMCQA		Financial-Phrasebank		SciQ		Social-i-QA	
	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$
LLaMA2-7B										
Zero-shot	4.6	4.6	4.2	4.2	34.1	34.1	8.0	8.0	41.1	41.1
EmbSim	43.6	51.6	33.0	34.0	65.0	64.7	65.6	72.0	49.1	42.3
GraphSim	**47.8**	**53.4**	**35.0**	**35.0**	**68.5**	**73.3**	**68.8**	**74.4**	**51.3**	**42.5**
$\mathcal{L}_{LLM}$	45.6	50.4	31.0	34.2	52.8	63.7	64.0	74.2	41.5	50.1
GLIP	**46.6**	**50.8**	**33.0**	**35.2**	**56.3**	**68.9**	**66.6**	**75.0**	**41.5**	**51.7**
Ours	46.2	50.4	32.6	34.6	54.8	66.9	65.6	74.5	41.3	51.3
Oracle	48.0	51.2	34.0	36.2	56.5	70.9	66.6	75.2	41.7	52.7
LLaMA2-13B										
Zero-shot	52.0	52.0	9.2	9.2	65.4	65.4	55.8	55.8	55.3	55.3
EmbSim	59.2	66.0	39.0	21.6	77.2	76.6	83.4	84.6	63.7	49.1
GraphSim	**62.6**	**66.4**	**39.4**	**25.6**	**79.4**	**83.0**	**84.8**	**84.8**	**64.5**	**53.9**
$\mathcal{L}_{LLM}$	62.6	66.2	38.0	38.6	55.3	77.6	82.6	85.8	62.3	63.7
GLIP	**63.8**	**67.2**	**39.8**	**41.4**	**72.4**	**86.6**	83.2	86.2	**62.7**	63.7
Ours	63.4	67.2	38.6	41.0	71.4	86.2	**83.2**	**86.2**	62.5	**63.7**
Oracle	64.6	67.6	41.0	43.0	71.6	87.2	84.4	87.0	62.7	63.8
GPT-3.5										
Zero-shot	74.6	74.6	49.6	49.6	57.5	57.5	91.2	91.2	74.0	74.0
EmbSim	78.2	**81.2**	50.0	53.2	81.2	93.6	92.2	**94.2**	74.2	72.4
GraphSim	**81.6**	79.6	**55.2**	**53.6**	**93.6**	**94.4**	**95.6**	94.0	**74.4**	**72.6**
$\mathcal{L}_{LLM}$	82.0	80.8	58.8	60.8	76.4	76.0	94.4	94.4	72.0	**76.0**
GLIP	**83.2**	**84.0**	**60.0**	**63.2**	**83.2**	**78.4**	95.2	**94.4**	**75.2**	74.8
Ours	82.6	82.6	59.2	62.0	81.2	77.2	**95.4**	94.2	73.4	74.6
Oracle	82.4	83.2	60.4	61.6	85.6	91.2	96.8	96.4	73.6	77.2

5.3 Ablation Study of GraphSim Components

We conduct an ablation study to assess the individual contributions of GraphSim's key components to overall performance. Specifically, we evaluate GraphSim on two tasks: ARC-Challenge and Social-i-QA, using LLaMA2-13B under the one-shot in-context learning setting. As shown in Figure 3, we examine the effects of removing (i) adjacency-based aggregation (GraphSim w/o Adj) and (ii) GNN-based aggregation (GraphSim w/o GNN). Note that removing both components reduces GraphSim to EmbSim.

The results show that excluding GNN-based aggregation leads to a significant performance drop across both tasks, underscoring the importance of capturing multiple structural perspectives for robust cross-task example selection. Overall, both components contribute meaningfully to the effectiveness of GraphSim. In Section 6, we provide an empirical analysis of the influence of the number of GNNs in GNN-based aggregation

5.4 Influence of Number of ICL Examples

In this subsection, we investigate how the number of in-context examples affects ICL performance.

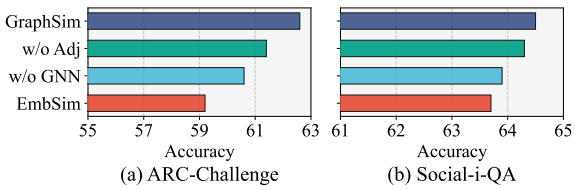


Figure 3: Ablation Study of GraphSim Components.

Specifically, we compare our pipeline with two cross-task ICL baselines: EmbSim and GraphSim, and one in-task ICL baseline $\mathcal{L}_{\text{LLM}}$. Specifically, we evaluate methods on two tasks: Financial-Phrasebank and MedMCQA, using LLaMA2-7B under the different shot ICL setting. The results are presented in Figure 4.

We observe that while cross-task ICL methods may perform well in low-shot settings (e.g., one-shot or four-shot), their performance degrades as the number of examples increases. This decline is likely due to distribution shifts and label space mismatches between source and target tasks. Adding more cross-task examples introduces noise that can confuse the LLM. This observation is consistent with the findings of Chatterjee et al. (2024). Nevertheless, GraphSim demonstrates greater robustness than EmbSim, due to its incorporation of structural

Table 4: Total runtime (in minutes) for pseudo-labeling and inference on 500 MedMCQA test samples using LLaMA2-7B. We vary the sizes of $\mathcal{D}^L$ and $\mathcal{D}^U$; for example, "100/400" means $|\mathcal{D}^L| = 100$, $|\mathcal{D}^U| = 400$.

Method	100/400	200/400	400/800	100/1600
GLIP	3.0	3.0	3.2	3.5
Ours	3.4	3.8	4.3	4.9
$\mathcal{L}_{\text{LLM}}$	3.6	3.6	5.3	8.4

information that generalizes better across tasks.

On the other hand, in-task ICL methods benefit from an increasing number of examples. Our pipeline achieves performance comparable to $\mathcal{L}_{\text{LLM}}$, while being more cost-efficient, as it relies less on expensive LLM calls.

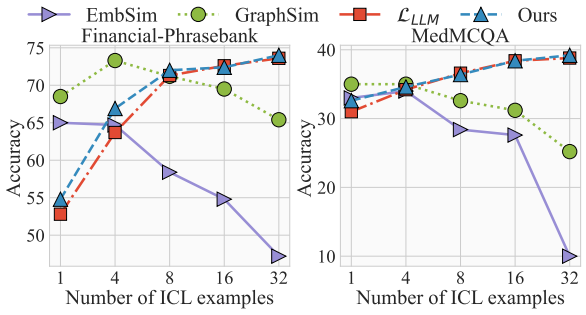


Figure 4: Accuracy variation with respect to the number of ICL examples using LLaMA2-7B.

5.5 Discussion of Labeling Efficiency

In this section, we analyze the labeling efficiency of different methods. Specifically, we measure the total runtime for pseudo-labeling and inference on 500 test samples from the MedMCQA task using the LLaMA2-7B model on an NVIDIA A6000 GPU. Results are presented in Table 4.

We observe that the runtime of $\mathcal{L}_{\text{LLM}}$ increases approximately linearly with the size of $\mathcal{D}^U$, as each additional sample requires an LLM call. This results in a significant increase in overall cost, especially when using commercial API-based LLMs,and the issue is further exacerbated with larger models due to their higher inference latency.

While our full pipeline introduces some overhead compared to GLIP alone (due to the additional GraphSim step for pseudo-labeling $\mathcal{D}^L$), it only requires LLM calls on a small subset of samples. As a result, the total cost grows much more slowly with the size of $\mathcal{D}^U$, making our method a more efficient and scalable solution for pseudo-labeling in practice.

Table 5: Performance of GraphSim under the one-shot setting using LLaMA2-13B. We vary the number and depth of GNNs; for example, "[1,1,2,2]" indicates two GNNs with one layer and two with two layers.

GNN Setting	ARC-Challenge	MedMCQA	SciQ
[1,2]	61.2	**39.6**	84.6
[1,1,2,2]	**62.6**	39.4	**84.8**
[1,1,1,2,2,2]	61.6	38.6	84.2

6 Ablation Study of Different Number of GNNs in GraphSim

We conduct ablation experiments to analyze how the number of GNNs in GraphSim affects performance. The evaluation is carried out on three target tasks under one-shot settings using the LLaMA2-13B model. Results are shown in Table 5.

We observe that using four GNNs, two with a single layer and two with two layers, yields the best average performance. Using fewer GNNs reduces the model's ability to capture structural information from diverse perspectives. On the other hand, increasing the number of GNNs leads to performance degradation due to the curse of dimensionality, as the final embedding size grows linearly with the number of GNNs, making similarity computation less meaningful in high-dimensional space. These results suggest that selecting an appropriate number of GNNs is crucial for balancing expressiveness and representational efficiency in cross-task similarity computation.

7 Conclusions

We propose a cost-effective and scalable pipeline for pseudo-labeling novel target tasks via cross-task ICL. we introduce GraphSim, a graph-based method that incorporates structural information to improve cross-task similarity estimation. To further reduce LLM reliance, we develop GLIP, a graph-based label propagation technique that extends a small pseudo-labeled seed set to the full target dataset without additional LLM queries. Our approach combines the generalizability of cross-task supervision with the efficiency of structure-aware label propagation. Experiments across multiple NLP benchmarks show that our method achieves competitive ICL performance while significantly lowering annotation and inference costs. We hope this work encourages future research on graph-enhanced ICL and promotes practical solutions for applying LLMs in low-resource scenarios.

8 Limitations

While our work provides an efficient pipeline for cross-task pseudo-labeling to facilitate in-context learning on novel tasks, there remain several promising avenues for future research.

- *Focus on Cross-Task Pseudo-Labeling*: Our goal is to construct demonstration pools for unseen target tasks by leveraging labeled examples from well-established source tasks. The current framework assumes access to one suitable source task, but identifying the best source task for a given target remains an open question. A promising direction is to measure the similarity between source and target task definitions, for instance, using the final-layer hidden representations of their prompts (Chatterjee et al., 2024). We leave a systematic study of source task selection strategies to future work.

- *Applicability Beyond Multiple-Choice QA*: We primarily focus on multiple-choice tasks in this work. Although certain generative or reasoning tasks can be reformulated into multiple-choice formats—e.g., by prompting an LLM to generate plausible distractors—such transformations introduce additional cost and complexity. Investigating the effectiveness and efficiency of our pipeline in these reformulated settings is left for future work.

- *Mixed-Task In-Context Learning*: Prior works such as Chatterjee et al. (2024) explore mixed-task ICL, where demonstrations are drawn from multiple tasks. Our study focuses instead on the single-source-task setting to investigate how to best exploit one available source for cross-task labeling. Extending our method to handle multi-source or mixed-task scenarios is an important direction for future research.

9 Ethics Statement

Our work focuses on developing a cost-efficient cross-task ICL labeling pipeline using publicly available datasets and pretrained language models. While acknowledging the need for responsible usage of the proposed method, we do not foresee major negative societal impacts.

Acknowledgments

This work is supported in part by the National Science Foundation (NSF) under grants IIS-2006844, IIS-2144209, IIS-2223769, CNS-2154962, BCS-2228534, CMMI-2411248, ECCS-2143559, and CPS-2313110; the Office of Naval Research (ONR) under grant N000142412636; and the Commonwealth Cyber Initiative (CCI) under grant VV-1Q24-011.

References

Rishabh Agarwal, Avi Singh, Lei Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, et al. 2024. Many-shot in-context learning. *Advances in Neural Information Processing Systems*, 37:76930–76966.

AI@Meta. 2024. Llama 3 model card.

Sören Auer, Dante A. C. Barone, Cassiano Bartz, Eduardo G. Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, Ivan Shilin, Markus Stocker, and Eleni Tsalapati. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports*, 13(1):7240.

Jinheon Baek, Sun Jae Lee, Prakhar Gupta, Siddharth Dalmia, Prateek Kolhar, et al. 2024. Revisiting in-context learning with long context language models. *arXiv preprint arXiv:2412.16926*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Anwoy Chatterjee, Eshaan Tanwar, Subhabrata Dutta, and Tanmoy Chakraborty. 2024. Language models can exploit cross-task in-context learning for data-scarce novel tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11568–11587.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023. Self-icl: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662.

Zihan Chen, Song Wang, Cong Shen, and Jundong Li. 2024. Fastgas: Fast graph-based annotation selection for in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9764–9780.

Zihan Chen, Song Wang, Zhen Tan, Jundong Li, and Cong Shen. Maple: Many-shot adaptive pseudo-labeling for in-context learning. In *Forty-second International Conference on Machine Learning*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*.

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. *arXiv preprint arXiv:2106.09685*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. *arXiv preprint arXiv:2306.01200*.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2022. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Jinhyuk Lee, Anthony Chen, Zhuyun Dai, Dheeru Dua, Devendra Singh Sachan, Michael Boratko, Yi Luan, Sébastien MR Arnold, Vincent Perot, Siddharth Dalmia, et al. 2024. Can long-context language models subsume retrieval, rag, sql, and more? *arXiv preprint arXiv:2406.13121*.

Zhenyu Lei, Yushun Dong, Weiyu Li, Rong Ding, Qi Wang, and Jundong Li. 2025a. Harnessing large language models for disaster management: A survey. *arXiv preprint arXiv:2501.06932*.

Zhenyu Lei, Zhen Tan, Song Wang, Yaochen Zhu, Zihan Chen, Yushun Dong, and Jundong Li. 2025b. Learning from diverse reasoning paths with routing and collaboration. In *The 2025 Conference on Empirical Methods in Natural Language Processing*.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. Mtop: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962.

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. In *NeurIPS*.

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*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65.

Marie Mikulová, Milan Straka, Jan Štěpánek, Barbora Štěpánková, and Jan Hajič. 2023. Quality and efficiency of manual annotation: Pre-annotation bias. *arXiv preprint arXiv:2306.09307*.

OpenAI. 2023. Introducing GPT-3.5 Turbo. `https://openai.com/blog/gpt-3-5-turbo`. Accessed: date-of-access.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference*

on Health, Inference, and Learning, volume 174 of *Proceedings of Machine Learning Research*, pages 248–260. PMLR.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1150–1160.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728.

Lakshay Sharma, Laura Graesser, Nikita Nangia, and Utku Evci. 2019. Natural language understanding with the quora question pairs dataset. *CoRR*, abs/1907.01041.

Chengshuai Shi, Kun Yang, Zihan Chen, Jundong Li, Jing Yang, and Cong Shen. 2024. Efficient prompt optimization through the lens of best arm identification. *Advances in Neural Information Processing Systems*, 37:99646–99685.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022a. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Jianlin Su, Mingren Zhu, Ahmed Murtadha, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2022b. Zlpr: A novel loss for multi-label classification. *arXiv preprint arXiv:2208.02955*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Eshaan Tanwar, Subhabrata Dutta, Manish Borthakur, and Tanmoy Chakraborty. 2023. Multilingual llms are better cross-lingual in-context learners with alignment. *arXiv preprint arXiv:2305.05940*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Mengting Wan, Tara Safavi, Sujay Kumar Jauhar, Yujin Kim, Scott Counts, Jennifer Neville, Siddharth Suri, Chirag Shah, Ryen W White, Longqi Yang, et al. 2024. Tnt-llm: Text mining at scale with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5836–5847.

Song Wang, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. 2024a. Mixture of demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 37:88091–88116.

Xinru Wang, Hannah Kim, Sajjadur Rahman, Kushan Mitra, and Zhengjie Miao. 2024b. Human-llm collaborative annotation through effective verification of llm labels. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024c. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Jinbin Zhang, Nasib Ullah, and Rohit Babbar. 2025. Large language model as a teacher for zero-shot tagging at extreme scales. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3465–3478.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Zhuosheng Zhang, Shuohang Wang, Yichong Xu, Yuwei Fang, Wenhao Yu, Yang Liu, Hai Zhao, Chenguang Zhu, and Michael Zeng. 2022. Task compass: Scaling multi-task pre-training with task prefix. *arXiv preprint arXiv:2210.06277*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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.

Anni Zou, Zhuosheng Zhang, Xiangru Tang, et al. 2023. Meta-cot: Generalizable chain-of-thought prompting in mixed-task scenarios with large language models.

A Dataset details

In this section, we provide detailed descriptions of the source and target datasets used in our experiments, following the setup of Chatterjee et al. (2024). We also include the task-specific instructions used in prompting for each dataset.

A.1 Source datasets

We have used the following datasets as source datasets:

ARC-Easy: ARC-Easy (Clark et al., 2018) is a subset of the ARC (AI2 Reasoning Challenge) dataset containing multiple-choice science questions for 3rd–9th grade students. Each question has four options, with one correct answer. We use the 2,251-question training set as the source dataset.

AG-news: AG-news (Zhang et al., 2015) is a news classification dataset with articles categorized into four classes: sports, business, technology, and world. We randomly sample 10K articles from the 120K training set for source examples.

BoolQ: BoolQ (Clark et al., 2019) is a reading comprehension dataset with yes/no questions based on associated passages. The 9,427 labeled question-passage pairs are used as source examples.

Commonsense-QA: Commonsense-QA (Talmor et al., 2019) is a multiple-choice QA dataset requiring commonsense reasoning. Each question has five options, one of which is correct. We sample source examples from the 9,740-question training set.

QQP: Quora Question Pairs (QQP) (Sharma et al., 2019) dataset is curated for the task of natural language understanding. This dataset consists of question pairs collected from the popular question-answering website *Quora*, and the task is to determine if the questions are duplicates of each other. We sample 10K labeled pairs for source usage.

RACE: RACE (Lai et al., 2017) is a reading comprehension dataset sourced from English exams for students aged 12–18, with multiple-choice questions based on passages. We sample 10K passage-question pairs from the 87.9K available in the training set.

SST2: SST2 (Socher et al., 2013) is a sentiment classification dataset from the Stanford Sentiment Treebank, where each movie review is labeled as positive or negative. We use 10K samples from the 67.3K training examples.

A.2 Target datasets

We have used the following datasets as target datasets:

ARC-Challenge: ARC-Challenge (Clark et al., 2018) is a subset of the ARC (AI2 Reasoning Challenge) dataset containing difficult science questions for students in grades 3–9. These are questions that were incorrectly answered by both a retrieval-based system and a word co-occurrence method, making them more challenging. Each question is multiple-choice with four options, one of which is correct. We randomly select 500 questions from the 1,172-question test set for our target dataset.

Social-i-QA: Social-i-QA (Sap et al., 2019) is a commonsense reasoning dataset focused on social and emotional understanding. Each example presents a social scenario and a multiple-choice question with three options. We sample 500 examples from the 1,954 available in the validation set for use as our target data.

SciQ: SciQ (Auer et al., 2023) is a multiple-choice science QA dataset covering topics in physics, chemistry, and biology. Each question has four answer choices. We construct our target dataset by sampling 500 examples from the 1,000-question test set.

MedMCQA: MedMCQA (Pal et al., 2022) is a multiple-choice QA dataset derived from Indian postgraduate medical entrance exam questions. Each question has four options, with one correct answer. We sample 500 questions from the 4,183-question validation set to form our target dataset.

Financial-Phrasebank: Financial-Phrasebank (Malo et al., 2014) is a sentiment analysis dataset consisting of 4,840 sentences from English-language financial news, categorized by sentiment.

Table 6: Additional experiment results using two recent LLMs, LLaMA3-8B and GPT-4o. Dashed lines separate cross-task ICL results from in-task ICL results. Bold font highlights the best performance within each category, while underlined values indicate the second-best results within the in-task ICL group for clearer analysis.

Method	ARC-Challenge		MedMCQA		Financial-Phrasebank		SciQ		Social-i-QA	
	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$	$K=1$	$K=4$
LLaMA3-8B										
Zero-shot	6.4	6.4	3.2	3.2	73.2	73.2	2.4	2.4	4.8	4.8
EmbSim	80.0	79.2	57.6	56.2	69.6	71.8	90.4	90.8	72.8	65.2
GraphSim	**81.0**	**80.0**	**60.0**	**57.2**	**72.4**	**73.2**	**90.6**	**92.8**	**73.2**	**72.4**
$\mathcal{L}_{LLM}$	78.4	80.4	57.2	60.0	82.2	**86.8**	89.6	91.6	71.2	**72.4**
GLIP	79.6	**82.4**	**57.2**	60.0	**82.4**	86.4	**92.4**	92.4	**71.4**	71.2
Ours	**80.0**	81.2	56.6	**60.0**	82.0	86.6	91.8	**92.4**	70.6	71.4
Oracle	81.6	82.0	57.6	59.6	86.0	92.4	92.8	93.6	73.2	71.6
GPT-4o										
Zero-shot	93.6	93.6	73.2	73.2	71.6	71.6	98.4	98.4	82.8	82.8
EmbSim	92.0	93.2	73.8	72.6	86.3	**89.2**	98.2	97.6	**81.2**	80.3
GraphSim	**94.2**	**94.4**	**74.4**	**73.5**	**88.6**	88.9	**98.4**	**97.9**	80.9	**82.1**
$\mathcal{L}_{LLM}$	96.0	96.0	**75.2**	78.4	93.6	**92.8**	96.6	98.8	**81.6**	82.4
GLIP	**97.2**	**96.6**	74.8	**78.8**	**94.0**	92.2	**98.4**	**98.8**	80.8	**83.6**
Ours	96.2	95.2	75.0	78.0	92.8	91.8	97.5	97.9	80.3	83.0
Oracle	95.6	96.4	77.6	78.4	96.8	96.4	98.0	98.8	83.4	84.2

The annotators were instructed to assess the sentences from an investor's perspective, determining whether the news would likely have a positive, negative, or neutral impact on stock prices.

In each case, the selection, though random, is done in such a way that our target datasets are balanced, i.e. the number of examples with each of the different possible labels is almost equal.

B Additional Experiments on More LLMs

In this section, we evaluate our pipeline on two additional large language models: LLaMA3-8B (AI@Meta, 2024) and GPT-4o (Hurst et al., 2024). Note that these models were not included in the experiments of Chatterjee et al. (2024). For cross-task ICL, we pair LLaMA3-8B with the same source-target settings used for LLaMA2-7B, and use the same GPT-3.5 settings for GPT-4o. While this may not yield the optimal source task selection for each model, it still allows us to analyze overall performance trends.

The results, shown in Table 6, indicate that GraphSim consistently outperforms EmbSim across different tasks, reaffirming its effectiveness for cross-task example selection. In the in-task setting, we observe that $\mathcal{L}_{LLM}$ achieves performance comparable to our pipeline when using GPT-4o. This is largely due to the strong capabilities of GPT-4o; however, $\mathcal{L}_{LLM}$ also incurs higher labeling costs. Overall, our pipeline continues to provide a cost-efficient solution for generating high-quality pseudo-labels on novel target tasks, even when compared to stronger LLMs.

Table 7: Comparison with demonstration-generation ICL baselines.

Method	MedMCQA	Financial-Phrasebank	Social-i-QA
Self-ICL	11.2	5.2	27.6
SG-ICL	15.4	6.2	28.0
Ours	34.6	66.9	51.3

C Additional Comparison with Demonstration-generation ICL Methods.

In addition to cross-task examples and zero-shot prompting, recent research has explored prompting LLMs to generate pseudo-demonstrations for a given task, which are then used for ICL at test time. In this section, we compare our proposed approach with two domain-agnostic demonstration-generation methods: Self-ICL (Chen et al., 2023) and SG-ICL (Kim et al., 2022). We use LLaMA2-7B as the base LLM in a 4-shot setting, with results shown in Table 7.

Both baseline methods show limited performance on the target tasks, primarily because the LLM struggles to understand the task from the description alone, consistent with the weak performance observed in the zero-shot ICL setting. As a result, these methods are unable to generate

Table 8: Influence of λ.

Dataset	0.2	0.4	0.6	0.8	1.0
Financial-Phrasebank	86.4	86.6	**86.8**	85.6	85.2
SciQ	85.4	**86.2**	86.0	85.8	85.0

high-quality demonstrations, highlighting the importance of better example selection strategies like ours.

D Ablation study.

In this section, we present an ablation study on the hyperparameter λ, which balances the training objective for the GNN. We experiment with λ values ranging from 0.2 to 1.0 on the Financial-Phrasebank and SciQ tasks. We report 4-shot performance using GLIP with LLaMA2-13B as the base LLM. The results are shown in Table 8.

Performance remains relatively stable across different values of λ, demonstrating the robustness of our method to this hyperparameter. While tuning λ on a development set could yield marginal improvements, it would also require substantially more LLM queries. To balance performance and computational cost, we adopt a fixed value that already delivers strong results.

Source task	Task definition
AG-news	Given a sentence do text classification, the sentence is a clipping from a news article that may be either related to sports, business, technology, or world news. You are to recognize the category of the sentence and label them as "sports", "business", "technology" or "world" news
ARC-Easy	Given a question answering task from the 3rd to 9th-grade science exam. The question contains four options "A.", "B.", "C." and "D." Select the most appropriate choice that answers the question
BoolQ	Given a context and a question do binary true and false type text classification. You are given a passage as context and a question related to the passage that can be answered as "True" or "False". Based on the context, question and your reasoning ability answer in a "True" and "False".
Commonsense-QA	The following task relates to commonsense reasoning. It consists of a question that can be easily solved using logical abilities and reasoning, a set of five options "A.", "B.", "C.", "D." and "E." are also provided along with the question, one of these options answers the question logically. Use your reasoning ability to select the most appropriate answer from the provided choices "A.", "B.", "C.", "D." and "E." and assign these choices (i.e "A.", "B.", "C.", "D." and "E.") as the label
QQP	Given two question pairs do text classification based on whether they are duplicates or not. The questions are mined from the popular online discussion forum Quora. As duplicate quetion might be present on Quora, the task is to label two identical questions as "duplicate" if they ask the same query else label the pair as "not duplicate".
RACE	Given a reading comprehension type question-answering from an english exam for school students. You are given a context and multiple choice question containing four options "A.", "B.", "C." and "D.". The question is answerable from the comprehension. Based on the question, the option and the context select the most appropriate answer from the provided choices "A.", "B.", "C." and "D.".
SST2	Given a movie review do text classification, based on the sentiment conveyed by the review label it as "positive" or "negative"

Table 9: Task definitions of source tasks

Target task	Task definition
ARC-Challenge	Given a question answering task from the 3rd to 9th-grade science exam. The question contains four options "A.", "B.", "C." and "D." Select the most appropriate choice that answers the question
Financial-Phrasebank	Given a sentence mined from a financial news article, you are to determine the sentiment polarity of the sentence. The task deals with financial sentiment analysis. Based on the sentiment conveyed by the sentence, label the sentence as "negative", "positive" or "neutral"
MedMCQA	Given a multiple choice question containing four options "A.", "B.", "C." and "D." from a medical entrance exam. The question is related to a sub-field of medical science like Microbiology, Radiology, Ophthalmology, Surgery, Human anatomy, etc. Based on the question, the option and your knowledge of the medical field select the most appropriate answer from the provided choices "A.", "B.", "C." and "D.".
SciQ	Given a question from a scientific exam about Physics, Chemistry, and Biology, among others. The question is in multiple choice format with four answer options "A.", "B.", "C." and "D.". Using your knowledge about the scientific fields answer the question and provide the label "A", "B", "C" and "D" as answer
Social-i-QA	Given an action as the context and a related question, you are to answer the question based on the context using your social intelligence. The question is of multiple choice form with three options "A", "B" and "C". Select the most appropriate answer from the provided choices "A", "B" and "C".

Table 10: Task definitions of target tasks