In-Context Transfer Learning: Demonstration Synthesis by Transferring Similar Tasks

Anonymous ACL submission

Abstract

Considering the high cost of labeling demonstrations for in-context learning (ICL), many works propose synthesizing demonstrations from scratch using LLMs. However, the quality of the demonstrations synthesized from scratch is limited by the capabilities and knowledge of LLMs themselves. To address this, inspired by transfer learning, we propose In-Context Transfer Learning (ICTL), which synthesizes target task demonstrations by transferring labeled demonstrations from similar source tasks. ICTL consists of two steps: source sampling and target transfer. First, we define an optimization objective, which minimizes transfer error to sample source demonstrations similar to the target task. Then, we employ LLMs to transfer the sampled source demonstrations to the target task, matching the definition and format of the target task. Experiments on Super-NI show that ICTL outperforms synthesis from scratch by 2.0% on average, demonstrating the effectiveness of our method[1].

1 Introduction

In-context learning (ICL) is an effective approach for large language models (LLMs) to adapt to various tasks based on the brilliant generalization ability of LLMs (Xun et al., 2017; Song et al., 2023b; Luo et al., 2024a). During the inference with ICL, input not only includes user questions but also several demonstrations to guide LLMs in generating answers correctly. Considering the high cost of demonstration labeling, many methods utilize LLMs to synthesize demonstrations from scratch without human involvement (Kim et al., 2022; Jin and Lu, 2024; Wang et al., 2025a). For instance, Self-ICL (Chen et al., 2023b) employs LLMs to synthesize demonstrations based on the task definition, while Su et al. (2024) improves the synthesis quality through iterations, where each iteration uses the results of previous iterations.

[1]Our code and data will be released upon acceptance.

However, the synthesis using LLMs from scratch is *constrained by the capabilities and knowledge of LLMs*, limiting the quality of the synthesized demonstrations (Yu et al., 2023). For example, a model not trained on coding tasks cannot synthesize code demonstrations well (Rozière et al., 2024; Luo et al., 2024b). To solve this issue, thereby improving ICL performance while reducing human involvement, motivated by transfer learning (Pan and Yang, 2010; Iman et al., 2023), we ***propose to synthesize demonstrations for the target task by transferring the labeled demonstrations of similar tasks***. The idea of transfer learning is inspired by the fact that previous works show that given similar source tasks, the performance of the target task can be enhanced by learning from the similar source task (Sun et al., 2020; Wang et al., 2024c; Nam et al., 2024). For example, as shown in Figure 1, the model can combine the context and the input of the source demonstration as the synthesized demonstration, which is then used for the target task.

Based on the above discussion, we present **I**n-**C**ontext **T**ransfer **L**earning (ICTL), which obtains the demonstrations of the target task by transferring the demonstrations of the source tasks. ICTL consists of two steps: ***sample*** the demonstrations similar to the target task, and ***transfer*** the sampled demonstrations for the target task, as shown in Figure 1. First, we present an optimization objective to measure the transfer error, based on which we minimize the transfer error to sample the demonstrations highly similar to the target task. Then, taking the sampled results and the target task definition as the input, we transfer the sampled demonstrations to the target task with LLMs.

To validate ICTL, we conduct experiments on Super-NaturalInstructions (Super-NI) (Wang et al., 2022), which can fully evaluate the multi-task capability of models with more than $1,600$ different tasks. Compared to the demonstration synthesis by LLMs from scratch, our method achieves an

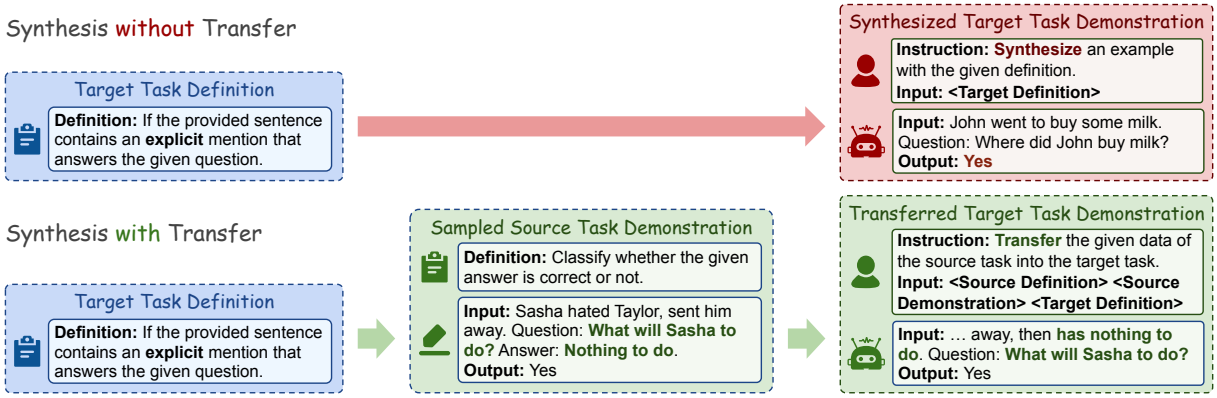


Figure 1: Comparison between previous demonstration synthesis methods (top) and our method (bottom). The blue part denotes the definition of the target task. The previous method synthesizes the demonstration from scratch, while the model misinterprets the definition and generates a demonstration with the wrong answer, where the answer is not *explicit* mentioned by the sentence. In contrast, our method synthesizes demonstrations by transferring the sampled demonstrations, reducing the capability reliance of LLMs. The corresponding parts between the source and the target demonstrations of our method are marked in **bold**.

average 2.0% performance improvement, demonstrating its effectiveness. We also validate ICTL on four math-reasoning and tool-use datasets, where it achieves an average relative improvement of 3.2% over the strongest baseline, demonstrating its effectiveness in practical application settings.

Our contributions are as follows:

- We propose synthesizing demonstrations by transferring labeled demonstrations of similar tasks, overcoming the constraint by the capabilities and knowledge of LLMs;
- We introduce an optimization objective to guide the source sampling, ensuring the similarity between the sampled results and the target task;
- Experiments on Super-NI show that, compared with the synthesis from scratch, ICTL delivers a 2.0% performance improvement on Super-NI, proving its effectiveness.

2 Methodology

In this section, we present ICTL, which synthesizes the demonstrations of the target task by transferring the labeled source demonstrations. The illustration of ICTL is shown in Figure 2, which consists of two steps: source sampling (§2.1) and target transfer (§2.2). Following the previous methods (Wang et al., 2024a; Yang et al., 2024), we synthesize demonstrations for each target task offline, where we do not synthesize for each target question since we want to ensure high efficiency of the inference. The prompts we used can be seen in Appendix B.1. The computational efficiency analysis of ICTL is shown in Appendix C.1.

2.1 Source Sampling

The source sampling step is designed to sample demonstrations that are highly similar to the target task from the labeled source demonstrations. In this paper, we define demonstration similarity as minimizing the target task error after transferring. We first present an optimization objective to guide the source demonstration sampling by minimizing the transfer error. Then, we discuss how to sample the source task demonstrations similar to the target task using our objective specifically, ensuring the quality of the synthesized results.

2.1.1 Optimization Objective for Source Sample

Supposing S and T represent the source and target tasks, respectively. To mitigate the embedding space gap, we employ the same embedding model to encode all demonstrations and task definitions. Let the embedding space be $\mathbb{R}^d$, and let $D_S = \{s_i | s_i \in \mathbb{R}^d\}$ denote the set of embedding vectors for all source task examples. We define the probability measure $\hat{\mu}_S(X) = \frac{|\{s_i \in D_S | s_i \in X\}|}{|D_S|}$ for any $X \subseteq \mathbb{R}^d$. The definition of $\hat{\mu}_T$ is analogous to the definition of $\hat{\mu}_S$.

Let $\epsilon(h)$ denote the task error of the hypothesis h, W be the Wasserstein distance (Rabin et al., 2012) measuring the divergence between two distributions, N denote the sample scale for each task, and φ be a negligible function. Redko et al. (2017) proves that the error upper bound of the transfer learning satisfies that:

$$\epsilon_T(h) \leq \epsilon_S(h) + W(\hat{\mu}_S, \hat{\mu}_T) + \varphi(N_S, N_T) \quad (1)$$

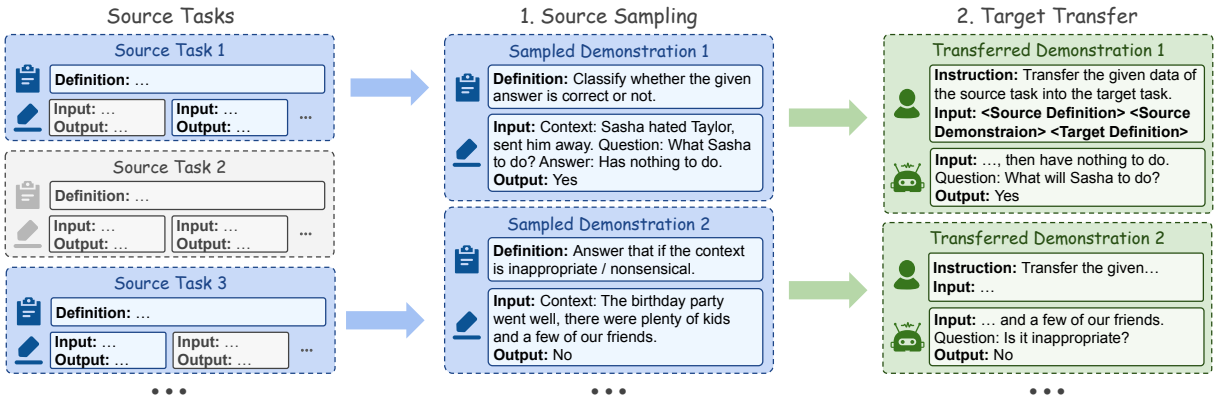


Figure 2: The illustration of ICTL, taking the target task definition "`If the provided sentence contains an explicit mention that answers the given question`" as an example. ICTL consists of two steps: *(i)* Source Sampling: sample demonstrations similar to the target task from the source tasks; *(ii)* Target Transfer: transfer the sampled demonstrations to the target task. The blue indicates the task definitions and demonstrations similar to the target task, and the gray part indicates the dissimilar. The green denotes the transferred demonstrations.

Further details of Equation 1 are discussed in Appendix A. From Equation 1, we can see that the upper bound of the error for the target task is mainly determined by the error of the source task and the divergence between the source and target tasks. It is hard to reduce the source task error since the source demonstrations cannot be modified. Therefore, we aim to optimize to minimize the target error by minimizing the divergence between the source tasks and the target task $W(\hat{\mu}_S, \hat{\mu}_T)$.

However, directly minimizing the upper bound results in $\hat{\mu}_T = \hat{\mu}_S$ could make the transferred demonstrations irrelevant to the target task x_T. Therefore, giving x as the representation vector of the task definition, we ask the target demonstrations $\hat{\mu}_T$ to satisfy that:

$$\hat{\mu}_T = \arg\min_{\hat{\mu}} W(\hat{\mu}, \hat{\mu}_S) + W(\hat{\mu}, x_T) \quad (2)$$

In Equation 2, the first term minimizes the divergence between the target and source demonstrations, ensuring similar knowledge and capability in the target task. The second term ensures that the target demonstrations are consistent with the target task definition. We discuss how to calculate the Wasserstein distance in Appendix B.2. We further discuss the effectiveness of Equation 2 with experiments in Appendix D.2.

Given a series of source tasks $\{S_i\}$, suppose N is the sampling scale of demonstrations from multiple source tasks $\{\hat{\mu}_{S_i}\}$, N_{S_i} is the sampled number of S_i and $\hat{\mu}$ is the empirical distribution of all possible sampled demonstrations across all source tasks. Based on Equation 1 and Equation 2, we can derive the following optimization objective to sample the source demonstrations:

$$\hat{\mu}_S = \arg\min_{\hat{\mu}} \sum_{S_i} \frac{N_{S_i}}{N} (6W(\hat{\mu}_{S_i}, x_T) + W(x_{S_i}, x_T)) \quad (3)$$

The proof of Equation 3 is provided in Appendix A. It can be observed that the first term in the summation ensures that the sampled source task demonstrations are similar to the target task definition, and the second term ensures that the definitions of the sampled source tasks are similar to the target task definition. Using Equation 3, we can sample demonstrations highly similar to the target task from the give source tasks, thereby lowering the error during the transfer and ensuring the quality of the transferred demonstrations.

2.1.2 Sampling with Equation 3

Based on Equation 3, we then discuss how to sample source demonstrations specifically. First, we embed the definitions and demonstrations of all source tasks, as well as the definition of the target task, into vectors using an embedding model. Following Wang et al. (2024c), we then filter the source tasks to select those most similar to the target task, reducing the overhead of subsequent calculations while ensuring performance. The filtering is done by ranking the Wasserstein distance between the embedding vectors of the source and target task definitions. From the filtered source tasks, we sample a fixed number of demonstrations using Equation 3. We employ a randomized algorithm for the source demonstration sampling, with details provided in the Appendix B.3.

2.2 Target Transfer

After sampling the source demonstrations, the target transfer step focuses on transferring the sampled demonstrations to the target task while ensuring that the transferred demonstrations are consistent with both the target task and the sampled demonstrations, transcending the limitations of the inherent capabilities and knowledge of LLMs. The target transfer consists of: *Transfer* and *Sample*.

Transfer The transfer step is designed to transfer the sampled demonstrations to match the target task definition and format. We employ LLMs for the transfer, where the input includes the definitions of both the source and target tasks, the source demonstration to be transferred, and a human-labeled example of the target task to specify the input and output formats. Furthermore, to ensure the format fidelity of the transferred results, we employ LLMs to verify that the transferred demonstrations conform to the format of the provided example. However, considering that LLMs are not reliable for assessing content accuracy (Wang et al., 2024b; Hu et al., 2024), we do not use LLMs to check whether the inputs and outputs of the transferred demonstrations are mutually consistent.

Sample The sample step aims to sample the transferred target demonstrations with Equation 2, ensuring that the sampled demonstration is consistent with the target task while staying similar to the sampled source demonstrations, thereby transcending the limitations of the capabilities and knowledge of LLMs. The sampling algorithm used for the transferred demonstration sampling is the same as the source sampling in §2.1.2, with the optimization objective defined by Equation 2. The sampled demonstrations are considered as the final output of our demonstration transfer method, which is then used for the inference of the target task.

3 Experiments

3.1 Experiment Setup

Dataset We use the Super-NaturalInstructions dataset (Super-NI) (Wang et al., 2022) to validate our method, which contains over $1,600$ tasks, allowing for a comprehensive evaluation of the cross-task generalization ability, demanding various knowledge and capabilities. To further validate the effectiveness of ICTL, we conduct additional experiments on two tasks: math reasoning (GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021)) and tool using (MetaTool (Huang et al., 2024), NesTools (Han et al., 2025)), which can reflect the practical effectiveness in real tasks. We introduce the above experimental datasets in detail in Appendix B.4.

Metric Following the Super-NI setup, we use Rouge-L (RougeL) and Exact Match (EM) as the evaluation metrics. RougeL measures the overlap between the predicted output and the reference answer, while EM assesses whether the predicted output exactly matches the reference.

Model We use BGE-EN-ICL (Chen et al., 2023a) to embed task definitions and demonstrations for the sampling. For the transfer and inference, we use Llama3.1-8b-Instruct (Llama3.1-8b) (Meta, 2024) and `GPT-4o` (OpenAI et al., 2024) as the experimental models, which are the mainstream LLMs. We mainly use Llama3.1-8b as the model of our analysis experiments due to the high cost of `GPT-4o`. About Llama3.1-8b, we use one single A100-80G, which costs about 2 hours for each running. For `GPT-4o`, it costs about $50\$$ for each running.

Baseline To thoroughly evaluate the effectiveness, we compare ICTL with the baselines:

- **Zero**: No demonstrations are provided during inference, using a zero-shot setting;
- **Direct**: Directly use the sampled source demonstrations without transferring;
- **Single**: Only use the single human-labeled example as the demonstration;
- **Synthesis**: Synthesize demonstrations from scratch based on the one example provided.

3.2 Main Experiment

As the result shown in Table 1, ICTL outperforms all baselines across different metrics and models in most categories, showing the effectiveness of our method across various settings. To further evaluate the effectiveness of our filter method, we compare it with other filter methods in Appendix D.1. Additionally, the results in the table also reveal that:

Baseline Notably, ICTL brings 2.0% improvement on average compared to the *Synthesis* setting. This shows that the demonstrations synthesized by LLMs from scratch are constrained by the capabilities and knowledge of LLMs themselves. In contrast, ICTL overcomes this constraint by providing the labeled demonstrations of other similar tasks, lowering the capability and knowledge requirement. Additionally, the *Direct* setting directly

Model	Category	Zero	Direct	Single	Synthesis	ICTL
Llama3.1-8b	Classification	62.5	60.3	61.9	65.4	**68.0**
	Comprehension	56.1	55.3	60.0	62.8	**67.8**
	Dialogue	57.2	62.7	65.2	**73.1**	72.3
	Extraction	43.4	38.7	48.3	**53.2**	51.2
	Generation	38.4	34.6	41.1	42.3	**45.8**
	Rewriting	46.6	32.6	58.1	60.5	**61.0**
	Overall (EM)	36.9	35.6	39.7	41.9	**44.0**
	Overall (RougeL)	52.0	48.8	54.7	57.8	**60.3**
GPT-4o	Classification	76.0	72.2	78.0	79.0	**81.0**
	Comprehension	78.4	76.4	74.9	72.2	**78.4**
	Dialogue	80.5	78.5	80.5	**83.5**	82.0
	Extraction	72.7	65.2	**73.0**	71.0	70.9
	Generation	39.1	38.4	42.6	44.5	**45.4**
	Rewriting	65.3	59.3	79.6	80.2	**80.7**
	Overall (EM)	49.2	44.6	49.4	49.7	**51.8**
	Overall (RougeL)	68.7	65.0	71.4	71.8	**73.1**

Table 1: Performance on Super-NI. The score on each specific category is RougeL. The best result of each category is highlighted in **bold**. Considering the cost, we only adapt experiments on 12 tasks of the Super-NI test set for GPT-4o, where we randomly select 2 tasks for each category, as shown in Appendix B.6.

using the sampled results as demonstrations leads to worse performance compared to the *Zero* setting. This indicates that transfer is necessary when using demonstrations from other tasks to enhance performance, even if the sampled source demonstrations are highly similar to the target task.

Task Based on the rates of different tasks shown in Appendix B.6, the performance improvement is more significant for tasks with a higher rate in the source tasks, as there are sufficient similar source demonstrations for transfer. However, our method slightly underperforms compared to other settings in the *Dialogue* and *Extraction* tasks. This is because these two tasks comprise only about 5% of the total data, leading to lower-quality transfer results due to a lack of similar source demonstrations. These findings suggest that it is important to use source demonstrations that are highly similar to the target task when using ICTL, as discussed in detail in §3.4.3. To better observe the transfer relationship between the source and target tasks of various categories, we state the transfer status of our method in Appendix D.3.

Metric Compared to EM, the performance improvement on RougeL is more significant. That is because EM is harder to improve since it requires the generated answer to be completely identical to the reference answer, while RougeL allows for partial matches and flexibility in answer formats, providing credit for partially correct outputs, making it relatively easier to improve.

Method	GSM8K $\rightarrow$ MATH	MATH $\rightarrow$ GSM8K	MetaTool $\rightarrow$ NesTools	NesTools $\rightarrow$ MetaTool
Zero	45.6	83.6	50.2	50.6
Direct	46.8	84.8	51.3	51.8
Single	47.0	85.2	51.1	52.4
Synthesis	47.2	85.6	50.9	53.2
ICTL	**48.2**	**87.2**	**52.2**	**57.0**

Table 2: EM on math and tool-use tasks using Llama3.1-8b. $A \rightarrow B$ denotes synthesizing demonstrations for the task B with the demonstrations of the task A. The best performance under each setting is marked in **bold**.

Method	EM	RougeL
ICTL	44.0	60.3
- Target Verify	41.4(−2.6)	56.3(−4.0)
- Source Sample	41.7(−2.3)	56.8(−3.5)
- Target Sample	43.7(−0.3)	60.0(−0.3)

Table 3: The ablation experiments on Super-NI using Llama3.1-8b. *(i)* Transfer Verify: remove target verification; *(ii)* Source Sample: sample source demonstrations randomly; *(iii)* Target Sample: use the verified demonstrations without sampling.

Model Besides, compared to Llama3.1-8b, the performance enhancement of GPT-4o is somewhat weaker. That is because that even under the *Zero* setting without demonstrations, GPT-4o is already capable of effectively addressing the tasks within Super-NI. Therefore, when the model struggles to tackle the target task adequately, our method can yield more significant performance gains.

To further validate the effectiveness of ICTL, we conduct experiments on two tasks: mathematical reasoning and tool using. The experimental results are shown in Table 2. As can be observed from the table, ICTL achieves 3.2% relative performance improvements on these two mainstream tasks, demonstrating the effectiveness of our approach. Moreover, it can be observed that transferring from relatively simple examples (e.g., GSM8K) can still improve performance on more challenging datasets (e.g., MATH). This occurs because LLMs could commit errors even on in-scope problems due to hallucination and limited robustness. By providing transferred demonstration highly aligned with the target task, ICTL mitigates hallucinations and enhances robustness.

3.3 Ablation Study

To verify the effectiveness of each component in ICTL, we conduct ablation studies of our method, where the experimental results are shown in Table 3. Based on the table, we analyze each ablation study in order of its impact on performance.

Target Verify Removing transfer verification results in the most significant performance drop of 3.3% on average across two metrics. This shows that the transferred demonstrations are inconsistent with the input and output format of the given example. There are two main reasons for the low inconsistency of demonstrations transferred directly: *(i)* For many test tasks, especially those that can be answered in multiple ways, it is difficult for LLMs to determine the format of the task, resulting in poor transfer results; *(ii)* Min et al. (2022) shows that LLMs could generate responses according to their prior experience during the pre-training while ignoring instructions, resulting in some generated demonstrations not meeting the task definition and format of the target task.

Source Sample Removing source sampling also causes a sharp performance drop of 2.9% on average. This is because, without source sampling, ICTL uses random sampling of source demonstrations, which leads to many dissimilar source demonstrations being sampled, decreasing the performance. This result proves the necessity of sampling the source demonstrations according to the similarity to the target task before the transfer. Besides, after removing source sampling, the performance of our method is near the *Synthesis* setting. This shows that when the source demonstrations provided are significantly different from the target task, LLMs are more inclined to synthesize results by themselves without referring to the demonstrations provided in the input.

Target Sample Removing target sampling has the least impact on performance, causing only a 0.3% decrease. This is because, considering that ensuring the similarity between the demonstration and the question can effectively ensure the performance of ICL (Shum et al., 2023; Yang et al., 2024), during the evaluation, we also select the demonstration corresponding to each question based on BM-25, which overlaps with the transfer sampling of ICTL to a certain extent.

3.4 Analysis

In this part, we analyze how different parameters affect the performance of ICTL to guide the selection of parameters in practical applications. To better observe the performance changes brought about by ICTL with the change of different parameters, we use the *Single* setting as our baseline. We also present the case study in Appendix C.2 to discuss how our method transfers the given demonstrations for the target task, and evaluates the performance of our method under human-labeled target task demonstrations and cross-domain settings in Appendix D.5 and Appendix D.6.

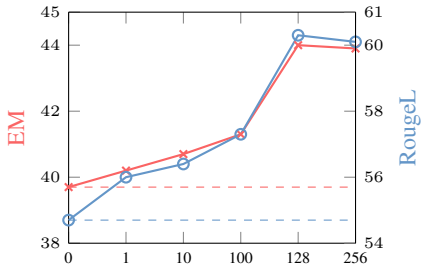
3.4.1 Source Demonstration Scale

The scale of source demonstrations available for different practical applications varies, so we analyze the impact of different scales of source demonstrations on the performance of our method, as shown in Figure 3a. From the figure, we can see that: *(i)* When the scale of the source demonstration sampling is smaller than 128, the overall experimental results exhibit an upward trend, demonstrating that increasing the amount of source demonstrations can effectively enhance the performance of our method; *(ii)* When the sampling scale exceeds 128, there is a slight decrease in performance, indicating that further addition of new source demonstrations does not continue to improve performance, as the number of demonstrations similar to the target task is limited. Therefore, when obtaining demonstrations of source tasks, it is necessary to obtain as many demonstrations as possible to ensure that there are enough different abilities or knowledge for the target task.

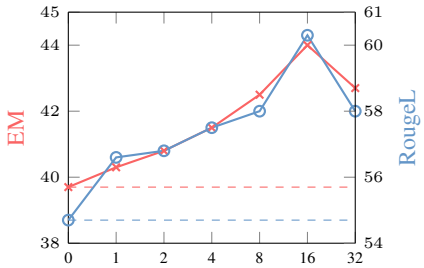
Notably, compared to not using transfer learning, even transferring using one source demonstration can also effectively improve the performance of the target task. This is because: *(i)* Even using one single source demonstration, we can also synthesize a large amount demonstrations of the target task, resulting in a high-quality demonstration pool and thus better performance than without transfer learning; *(ii)* Previous research (Kim et al., 2022; Wang et al., 2024a) and the *Synthesis* setting of Table 1 show that even without source demonstrations, LLMs can still synthesize demonstrations based on the inherent knowledge of themselves, thereby enhancing inference performance.

3.4.2 Source Task Scale

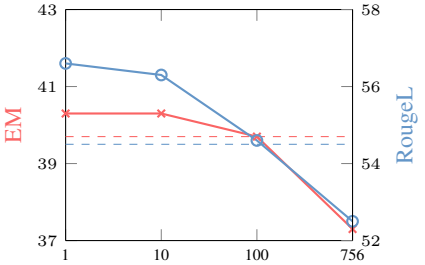
The scale of source tasks that can be obtained varies in practical applications, so we analyze the impact of different task scales on the performance of ICTL. The experimental results are shown in Figure 3b, from which we can see that: *(i)* When the scale of source tasks is less than 16, the overall performance exhibits an upward trend, while when the scale exceeds 16, the performance starts to decline sharply, showing that blindly increasing the scale



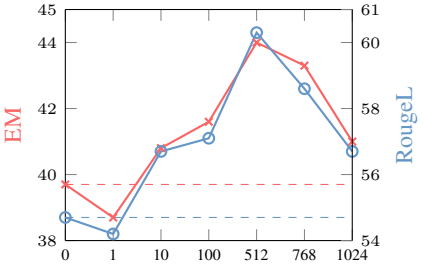
(a) The performance with demonstrations of different scales of the sampled source demonstrations.



(b) The performance with demonstrations of different scales of the sampled source tasks.



(c) The performance of demonstrations with different similarity ranks to the target task.



(d) The performance with demonstrations of different scales of the transferred target demonstrations.

Figure 3: The impact of different parameters on the performance of the Super-NI test set with ICTL using Llama3.1-8b. 0 of the X-axis indicates the performance under the *Single* setting. The red lines denote EM and blue lines denote RougeL. The dotted lines denote the performance of *Single* baseline.

of the source task cannot bring about continuous improvement and the importance of ensuring the similarity between the source and the target tasks; *(ii)* Compared to the source demonstration scale, the performance degradation is more pronounced with the increase in source task scale, since the scale of source tasks similar to the target task is limited, whereas simply increasing the scale of tasks, rather than the demonstrations, introduces more irrelevant information, leading to a more significant decrease in the quality of the transferred demonstrations and the inference performance.

3.4.3 Task Similarity Rank

Considering there could be many new tasks emerging in future research and applications, to explore the adaptability of ICTL to new tasks, we conduct experiments to examine the impact of the similarity between the source and target tasks on performance. We rank the Wasserstein distance of the embedding vectors of the source and target task definition in descending order, selecting the 1st, 10th, 100th, and last-ranked (756th in the Super-NI train set) source tasks to be transferred. The experimental results are shown in Figure 3c, from which we can observe the following: *(i)* When the similarity ranking of the source tasks is within the top 10, the performance of our method does not fluctuate significantly, since there exists multiple source tasks similar to those in the Super-NI test set, resulting in transferred demonstrations of comparable quality; *(ii)* After the similarity ranking exceeds 10, the performance of ICTL begins to decline sharply, indicating that demonstrations of tasks with large gaps can not help the target task, showing the importance of ensuring the task similarity.

3.4.4 Target Transfer Scale

Due to the computational resource limitation in practical applications, the scale of the transferred demonstrations could be limited. Therefore, we evaluate the performance of ICTL under different scales of transferred demonstrations, as shown in Figure 3d. From the figure, we can observe the following: *(i)* In cases where only one single demonstration is transferred, the model performance decreases compared to without transfer, since the quality of the single transferred demonstration is lower than the provided example labeled by humans, leading to a performance decline; *(ii)* Even only obtains 10 demonstrations by transferring, our method achieves better performance than no transfer, whereas the scale of transferred demonstrations increases, the performance improves accordingly, demonstrating the necessity of sufficient transferring; *(iii)* However, after the transferred

demonstrations reach a certain scale, the model performance plateaus, since the information contained in the sampled source demonstrations is fully represented with 512 transferred demonstrations, and further increasing the scale does not yield new high-quality demonstrations, lowering the performance since mixing low-quality demonstrations.

4 Related Works

4.1 Demonstration Synthesis

Demonstrations are of great importance in ICL, which can effectively help LLMs adapt to various target tasks (Brown et al., 2020; Dong et al., 2024). Considering the high cost of human labeling, many methods present to synthesize demonstrations using LLMs from scratch, lowering the human involvement (Kim et al., 2022; Chang and Fosler-Lussier, 2023; Jin and Lu, 2024). Some methods focus on ensuring the correctness of the synthesized demonstrations, meeting the task definitions by filtering out low-quality synthesized results (Chen et al., 2023b; Su et al., 2024; Yang et al., 2024). Another type of method aims to increase the diversity of the synthesized demonstrations, creating demonstrations dissimilar to synthesized results (Zhang et al., 2023; Shum et al., 2023; Wang et al., 2024a). Recent studies further enrich synthesis: Wang et al. (2025b) propose a distillation-based framework that generates stable target-specific demonstrations, while task-agnostic pipelines promote diversity and consistency (Wang et al., 2025a; Kothapalli et al., 2025).

However, the demonstrations synthesized by the current methods are constrained by the knowledge and capabilities of LLMs themselves, limiting their performance on the tasks unseen in their pre-training (Yu et al., 2023). Although human-labeled demonstrations for new task scenarios can help LLMs generalize to these new tasks, labeling demonstrations for any new task or domain is costly (Wang et al., 2013). To address these issues, we present ICTL, which synthesizes demonstrations for new target scenarios by transferring labeled demonstrations of the source tasks similar to the target task, addressing the limitation of the knowledge and capabilities of LLMs.

4.2 Deep Transfer Learning

Transfer learning is a widely researched direction aimed at helping models acquire the ability to solve target tasks based on their existing capabilities from the source tasks (Pan and Yang, 2010; Zhuang et al., 2020). With the impressive performance demonstrated by deep learning methods, deep transfer learning has become an important approach within the field of transfer learning (Iman et al., 2023). Some methods focus on transferring and freezing model parameters to retain and learn features of different tasks (Scialom et al., 2022; Song et al., 2023a; Wang et al., 2023; Rostami et al., 2023; Du et al., 2024; Somerstep et al., 2025; Tahir et al., 2025). Other transfer learning methods enhance the performance from the data perspective, studying how to adjust the training sequence of tasks, mix source task data with target task data, or modify the source task format to improve transfer learning performance (Xu et al., 2023; Wang et al., 2024c; Madine, 2024; Ju et al., 2025).

However, current transfer learning methods rely on the labeled data of the target task and the model training, leading to the high cost of the adaption considering the high cost of labeling and LLM training. Therefore, in this paper, we present employing transfer learning to enhance ICL by synthesizing demonstrations using the labeled source demonstrations, lowering human involvement and training costs, meanwhile helping LLMs adapt to various target tasks with high performance.

5 Conclusion

In this paper, we mainly discuss how to synthesize ICL demonstrations with high quality. Motivated by transfer learning, we propose ICTL, which synthesizes the demonstrations of the target task by transferring the similar labeled demonstrations, addressing the constraint that synthesizing from scratch with LLMs is limited by the capabilities and knowledge of LLMs. We first present an optimization objective for sampling source demonstrations, aiming to minimize transfer errors by ensuring that sampled demonstrations are highly similar to the target task. Subsequently, we transfer the sampled demonstrations to the target task using LLMs without human involvement, taking the sampled results and the target task definition as the input. Experiments on Super-NI demonstrate that our method achieves an average improvement of 2.0% over demonstrations synthesized without transfer, validating its effectiveness. Moreover, on the math and tool-use tasks, ICTL achieves an average relative improvement of 3.2% over the baseline, demonstrating its practical utility.

Limitations

(i) We only adapt experiments on two models, where we will adapt experiments on more models in the future work. *(ii)* We only use one single source task demonstration for each synthesized demonstration of the target task, which limits the diversity, where we will discuss how to synthesize with multiple source task demonstrations for a single target task demonstration in the future research.

Ethics Statement

All datasets and models used in this paper are publicly available, and our usage follows their licenses and terms. We employ the LLM tools for coding and writing polishing.

References

Dimitris Bertsimas and John Tsitsiklis. 1993. Simulated annealing. *Statistical Science*, 8(1):10–15.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shuaichen Chang and Eric Fosler-Lussier. 2023. Selective demonstrations for cross-domain text-to-SQL. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14174–14189, Singapore. Association for Computational Linguistics.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2023a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2309.07597.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023b. 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, Singapore. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. *Preprint*, arXiv:2301.00234.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. 2024. Unlocking continual learning abilities in language models. *Preprint*, arXiv:2406.17245.

Han Han, Tong Zhu, Xiang Zhang, MengSong Wu, Xiong Hao, and Wenliang Chen. 2025. NesTools: A dataset for evaluating nested tool learning abilities of large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9824–9844, Abu Dhabi, UAE. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024. Are LLM-based evaluators confusing NLG quality criteria? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9530–9570, Bangkok, Thailand. Association for Computational Linguistics.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, and Lichao Sun. 2024. Metatool benchmark for large language models: Deciding whether to use tools and which to use. In *The Twelfth International Conference on Learning Representations*.

Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. 2023. A review of deep transfer learning and recent advancements. *Technologies*, 11(2).

Ziqi Jin and Wei Lu. 2024. Self-harmonized chain of thought. *Preprint*, arXiv:2409.04057.

Li Ju, Xingyi Yang, Qi Li, and Xinchao Wang. 2025. Graphbridge: Towards arbitrary transfer learning in gnns. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ArXiv:2502.19252.

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. *Preprint*, arXiv:2206.08082.

Vignesh Kothapalli, Hamed Firooz, and Maziar Sanjabi. 2025. Cot-icl lab: A synthetic framework for studying chain-of-thought learning from in-context demonstrations. *arXiv preprint arXiv:2502.15132*. ACL 2025.

Yibin Lei, Liang Ding, Yu Cao, Changtong Zan, Andrew Yates, and Dacheng Tao. 2023. Unsupervised dense retrieval with relevance-aware contrastive pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10932–10940, Toronto, Canada. Association for Computational Linguistics.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.ICL: Demonstration-retrieved in-context learning. In *R0-FoMo:Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024a. In-context learning with retrieved demonstrations for language models: A survey. *ArXiv*, abs/2401.11624.

Xianzhen Luo, Qingfu Zhu, Zhiming Zhang, Libo Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024b. Python is not always the best choice: Embracing multilingual program of thoughts. *Preprint*, arXiv:2402.10691.

Manas Madine. 2024. Bridging distribution gap via semantic rewriting with LLMs to enhance OOD robustness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 458–468, Bangkok, Thailand. Association for Computational Linguistics.

Meta. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jaehyun Nam, Woomin Song, Seong Hyeon Park, Jihoon Tack, Sukmin Yun, Jaehyung Kim, Kyu Hwan Oh, and Jinwoo Shin. 2024. Tabular transfer learning via prompting LLMs. In *First Conference on Language Modeling*.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. 2012. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision*, pages 435–446, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ievgen Redko, Amaury Habrard, and Marc Sebban. 2017. Theoretical analysis of domain adaptation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases*, pages 737–753, Cham. Springer International Publishing.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Mohammad Rostami, Digbalay Bose, Shrikanth Narayanan, and Aram Galstyan. 2023. Domain adaptation for sentiment analysis using robust internal representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11484–11498, Singapore. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, and 7 others. 2024. Code llama: Open foundation models for code. *Preprint*, arXiv:2308.12950.

Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6107–6122.

Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore. Association for Computational Linguistics.

Seamus Somerstep, Felipe Maia Polo, Moulinath Banerjee, Yaacov Ritov, Mikhail Yurochkin, and Yuekai Sun. 2025. A transfer learning framework for weak to strong generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. Poster.

Chenyang Song, Xu Han, Zheni Zeng, Kuai Li, Chen Chen, Zhiyuan Liu, Maosong Sun, and Tao Yang. 2023a. Conpet: Continual parameter-efficient tuning for large language models. *arXiv preprint arXiv:2309.14763*.

Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. 2023b. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Comput. Surv.*, 55(13s).

Yi Su, Yunpeng Tai, Yixin Ji, Juntao Li, Yan Bowen, and Min Zhang. 2024. Demonstration augmentation for zero-shot in-context learning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 14232–14244, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. {LAMAL}: {LA}nguage modeling is all you need for lifelong language learning. In *International Conference on Learning Representations*.

Javan Tahir, Surya Ganguli, and Grant Rotskoff. 2025. Features are fate: A theory of transfer learning in high-dimensional regression. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.

Aobo Wang, Cong Duy Hoang, and Min-Yen Kan. 2013. Perspectives on crowdsourcing annotations for natural language processing. *Lang. Resour. Eval.*, 47(1):9–31.

Dingzirui Wang, Longxu Dou, Xuanliang Zhang, Qingfu Zhu, and Wanxiang Che. 2024a. Improving demonstration diversity by human-free fusing for text-to-sql. *Preprint*, arXiv:2402.10663.

Dingzirui Wang, Xuanliang Zhang, Keyan Xu, Qingfu Zhu, Wanxiang Che, and Yang Deng. 2025a. V-synthesis: Task-agnostic synthesis of consistent and diverse in-context demonstrations from scratch via v-entropy. *Preprint*, arXiv:2506.23149.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024b. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Wuyuqing Wang, Erkun Yang, Zilan Zhou, and Cheng Deng. 2025b. In-context learning demonstration generation with text distillation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-25)*, pages 6433–6441.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. 2023. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10658–10671, Singapore. Association for Computational Linguistics.

Yifan Wang, Yafei Liu, Chufan Shi, Haoling Li, Chen Chen, Haonan Lu, and Yujiu Yang. 2024c. InsCL: A data-efficient continual learning paradigm for fine-tuning large language models with instructions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 663–677, Mexico City, Mexico. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, and 16 others. 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihao Xu, Xuan Tang, Yufei Shi, Jianfeng Zhang, Jian Yang, Mingsong Chen, and Xian Wei. 2023. Continual learning via manifold expansion replay. *arXiv preprint arXiv:2310.08038*.

Guangxu Xun, Xiaowei Jia, Vishrawas Gopalakrishnan, and Aidong Zhang. 2017. A survey on context learning. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):38–56.

Jinghan Yang, Shuming Ma, and Furu Wei. 2024. Auto-icl: In-context learning without human supervision. *Preprint*, arXiv:2311.09263.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023. Large language model as attributed training data generator: A tale of diversity and bias. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. Revisiting out-of-distribution robustness in NLP: Benchmarks, analysis, and LLMs evaluations. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Preprint*, arXiv:1911.02685.

A Proofs

In this section, we present the proof of Equation 3. The proof includes three parts. First, we discuss how to measure the transfer error when transferring across multiple source tasks. Next, we address how to measure the discrepancy between the source tasks and the target task, denoted as $W(\hat{\mu}_S, \hat{\mu}_T)$. Finally, we combine the existing results to derive Equation 3.

$$\epsilon_T(\hat{h}_{\boldsymbol{\alpha}}) \leq \min_h \epsilon_T(h) + c_1 + 2\sum_{i=1}^{N} \alpha_i \left(W(\hat{\mu}_{S_i}, \hat{\mu}_T) + \lambda_i + c_2\right) \tag{4}$$

Suppose $\boldsymbol{\alpha} = \{\alpha_i\}$ represents the proportion of each source task, c_1, c_2 is dependent on n, N_{S_i}, N_T, and $\lambda_i = \min_h(\epsilon_{S_i}(h) + \epsilon_T(h))$ denotes the joint error of each source task S_i. Based on Equation 1, the previous work (Redko et al., 2017) has proved that, for the transfer learning across multiple source tasks, the error satisfies Equation 4.

$$\hat{\mu}_S = \arg\min_{\{\hat{\mu}_{S_i}\}_N} \sum_{i=1}^{N} \alpha_i W(\hat{\mu}_{S_i}, \hat{\mu}_T) \tag{5}$$

To minimize the error, we aim to minimize the upper bound of the error. Since $\min_h \epsilon_T(h) \leq \sum_{i=1}^{N} \alpha_i \epsilon_T(h_{S_i})$, and $\sum_{i=1}^{N} \alpha_i \lambda_i \leq \sum_{i=1}^{N} \alpha_i \epsilon_T(h_{S_i}) + \alpha_i \epsilon_{S_i}(h)$, by replacing $\epsilon_T(h_{S_i})$ with Equation 1, and ignoring the terms related to the error of source tasks and constants unrelated to μ, we can obtain Equation 5. Equation 1 defines how to sample the target demonstrations given the source demonstrations. Then, we discuss the upper bound of the value of Equation 1, where we can adjust the source demonstrations to minimize the upper bound, thereby lowering the transfer error.

Theorem A.1. *Let x_S, x_T represent the representation vectors of the task definition of S and T. If*

$$\hat{\mu}_T = \arg\min_{\hat{\mu}} W(\hat{\mu}, \hat{\mu}_S) + W(\hat{\mu}, x_T),$$

then

$$W(\hat{\mu}_S, \hat{\mu}_T) \leq 6W(\hat{\mu}_S, x_T) + W(x_S, x_T).$$

Proof. Let $\hat{\mu}_{S,T}$ represent the empirical distribution of the subset sampled from X_S, which has the data most close to x_T. It is obvious that $W(\hat{\mu}_{S,T}, x_T) \leq W(\hat{\mu}_S, x_T)$.

Because $\hat{\mu}_T = \arg\min_{\hat{\mu}} W(\hat{\mu}, \hat{\mu}_S) + W(\hat{\mu}, x_T)$, we can get:

$$\begin{aligned}
&W(\hat{\mu}_T, \hat{\mu}_S) + W(\hat{\mu}_T, x_T) \\
&\leq W(\hat{\mu}_{S,T}, \hat{\mu}_S) + W(\hat{\mu}_{S,T}, x_T) \\
&\leq W(\hat{\mu}_{S,T}, \hat{\mu}_S) + W(\hat{\mu}_S, x_T) \\
&\leq W(\hat{\mu}_{S,T}, x_T) + 2W(\hat{\mu}_S, x_T) \\
&\leq W(\hat{\mu}_{S,T}, \hat{\mu}_T) + W(\hat{\mu}_T, x_T) + 2W(\hat{\mu}_S, x_T)
\end{aligned}$$

Erase $W(\hat{\mu}_T, x_T)$ on both sides of the unequal sign, we can get:

$$\begin{aligned}
&W(\hat{\mu}_T, \hat{\mu}_S) \\
&\leq W(\hat{\mu}_{S,T}, \hat{\mu}_T) + 2W(\hat{\mu}_S, x_T) \\
&\leq W(\hat{\mu}_{S,T}, x_T) + W(\hat{\mu}_T, x_T) + 2W(\hat{\mu}_S, x_T) \\
&\leq 3W(\hat{\mu}_S, x_T) + W(\hat{\mu}_T, x_T) + W(\hat{\mu}_T, \hat{\mu}_S) \\
&\leq 3W(\hat{\mu}_S, x_T) + W(\hat{\mu}_{S,T}, \hat{\mu}_S) + W(\hat{\mu}_{S,T}, x_T) \\
&\leq 5W(\hat{\mu}_S, x_T) + W(\hat{\mu}_{S,T}, x_T) + W(x_T, x_S) \\
&\leq 6W(\hat{\mu}_S, x_T) + W(x_T, x_S)
\end{aligned}$$

Thus, we conclude:

$$W(\hat{\mu}_T, \hat{\mu}_S) \leq 6W(\hat{\mu}_S, x_T) + W(x_T, x_S).$$

□

Theorem A.1 provides an upper bound for measuring the difference between the demonstrations of the target task and the source task in task transfer, based on the discrepancy between the task definitions of the source and target tasks. The reason this measurement holds is that the demonstrations for the target task are entirely transferred from the source demonstrations and the target task definition, meaning they can describe its characteristics. By substituting Theorem A.1 into Equation 5, we can derive Equation 3.

B Additional Information

B.1 Prompts of ICTL

The prompts we used in ICTL are shown in Table 4, Table 5 and Table 6.

B.2 Calculation of Wasserstein Distance

In our implementation, we approximate the sliced Wasserstein distance between two empirical distributions. First, high-dimensional samples from the source and target are linearly projected onto one or more directions collected in the matrix Θ (rows

The Prompt of Transfer of ICTL

Convert an example from Task A into an example for Task B, ensuring that both examples are consistent in terms of domain and knowledge. A sample for Task A is provided below. Please create a corresponding example for Task B, while maintaining the same domain and knowledge context.
The definition of Task A: `{task_a_definition}`
The definition of Task B: `{task_b_definition}`

—

For example, given the following example for Task A:
Input:
`{task_A_question_demo}`
Reason:
`{task_A_rationale_demo}`
Answer:
`{task_A_answer_demo}`

The corresponding example for Task B could be:
Input:
`{task_B_question_demo}`
Reason:
`{task_B_rationale_demo}`
Answer:
`{task_B_answer_demo}`

—

Based on the above example, please transfer the following example from Task A to Task B:
Input:
`{task_A_question}`
Answer:
`{task_A_answer}`

Your output format should be as follows:
Input:
<Converted input of Task B >
Reason:
<Explanation of the converted >
Answer:
<Converted answer of Task B >

Table 4: The prompt of transfer.

of Θ define directions), yielding one-dimensional samples for each direction. For a given direction, we estimate the 1-D W_2 by sorting the projected samples and constructing their empirical cumulative distributions on a common grid. We then compute the discrete first differences of these cumulative profiles and take the mean of the squared discrepancies between source and target. This quantity is a standard discrete surrogate of the integral formulation of W_2^2 in one dimension (which compares quantile or cumulative functions). Finally, when multiple projection directions are used, we aggregate the per-direction estimates by averaging them, which yields the sliced Wasserstein distance—i.e., the expectation of the 1-D Wasserstein distances over random linear projections—providing a computationally efficient proxy for comparing high-dimensional distributions.

During the calculation of the Wasserstein distance, if there is one input that is a single vector x, we consider it as the probability measure satisfying:

$$\mu_x(X) = \begin{cases} 1 & x \in X \\ 0 & x \notin X \end{cases} \tag{6}$$

When calculating the Wasserstein distance, if an input is a point (vector), we regard it as a distribution with a variance of 0.

B.3 Algorithm for Dataset Sampling

B.3.1 Sampling Algorithm of ICTL

In this part, we introduce the specific design of the randomized algorithm for sampling. The algorithm utilizes simulated annealing (Bertsimas and Tsitsiklis, 1993) to optimize the sampling of demonstrations most similar to the target task with low computational costs.

The Prompt of Verification of ICTL

Given a task description, several examples, and a pre-synthesized example, evaluate whether the pre-synthesized example matches the format and functionality of the provided examples and aligns with the task description. Based on the evaluation, determine whether the pre-synthesized example is "Qualified"
You should check the pre-synthesized example based on the following criteria:
1. Format Consistency: Does the pre-synthesized example follow the format of the provided examples?
2. Task Fulfillment: Does the pre-synthesized example fulfill the requirements of the task description?
3. Functional Accuracy: Are the input and output in the pre-synthesized example consistent with those in the provided examples?
If the pre-synthesized example meets all the criteria above, return: "Qualified."
If the pre-synthesized example fails to meet any of the criteria, return: "Unqualified."
Think it step by step.
 	
Task Description:
`{definition}`
 	
Examples:
 	
Input:
`{input_demo}`
Reason:
`{reason_demo}`
Answer:
`{answer_demo}`
 	
—
 	
...
 	
—
 	
Pre-synthesized Example:
Input:
`{input_transferred}`
Reason:
`{reason_transferred}`
Answer:
`{answer_transferred}`

Table 5: The prompt of verification.

Simulated annealing is a probabilistic global optimization algorithm that initially accepts suboptimal solutions at high temperatures to avoid local optima. As the temperature gradually decreases, the algorithm converges. The initial solution is generated through random sampling, where samples from the given demonstrations are randomly selected as the starting candidate solution. We use Equation 3 and Equation 2 as the score function to evaluate the quality of random sampling from the given demonstrations, where we calculate the Wasserstein distance following (Rostami et al., 2023).

During each iteration, the algorithm perturbs the current candidate solution to generate a new one. If the algorithm fails to find a better solution after several attempts, the perturbations are triggered to escape local optima. Whether the perturbed candidate is accepted depends on the difference in scores between the new and current solutions. Even if the new candidate is worse, there is a certain probability it is accepted. This probability decreases as the temperature drops, promoting sufficient search space exploration.

The annealing process starts with an initial temperature of 1.0, with a cooling rate of 0.99. The temperature decays after each iteration until it reaches the minimum value of 10^{-4}, at which point the algorithm stops. Additionally, we set a threshold: if no better solution is found after 100 iterations, large-step perturbations are applied. Although our method demands the additional cost for computing simulated annealing compared with the general ICL methods, these costs are offline, where our method has the same inference cost as other general ICL methods.

B.3.2 Comparison with Other Methods

Considering that the source task data sampling algorithm in ICTL is replaceable, we denote the total

The Prompt of Inference of ICTL

```
{task_definition}
Here are some demonstrations of the task:

—

Input:
{input_demo}
Reason:
{reason_demo}
Answer:
{answer_demo}

—

...

—

Based on the above demonstrations, please generate a response to the following question.
Your output format should be as follows:
Reason:
<Explanation of the answer >
Answer:
<Your answer >
Think it step by step.

Input:
{input_user}
```

Table 6: The prompt of inference.

Method	EM	Complexity
Random	40.1	$O(N)$
Random Optimize	42.3	$O(N \cdot M)$
Simulated Annealing	**44.0**	$O(\log_r(t) \cdot N)$

Table 7: The performance with different sampling methods of ICTL on SuperNI using Llama3.1-8b. The best result is marked in **bold**.

	0.9	0.99	0.999
$1e-4$	43.3	43.7	43.8
$1e-5$	43.5	43.5	43.5
$1e-6$	44.0	43.9	44.0
$1e-7$	43.9	44.0	44.2
$1e-8$	44.1	44.0	43.9

Table 8: The performance of ICTL under different sampling parameters. The row name denotes the minimum temperatures, and the column name denotes the cooling rate.

sampling size as N, the cooling rate of simulated annealing as r, and the minimum temperature as t. We replace the algorithm with the following alternatives for comparison:

- Random: Randomly sample and return N results.
- Random Optimize: Randomly sample results over M rounds and select the best round as the returned result. In the experiments below, we perform random sampling 100 times.

The performance and time complexity of different sampling algorithms are shown in Table 7. As shown in the table, ICTL outperforms other baselines even when using the more efficient Random Optimize algorithm, demonstrating the effectiveness of ICTL. Moreover, as the complexity of the sampling algorithm increases, the accuracy improves accordingly. Therefore, when transferring to new tasks, an appropriate sampling algorithm should be selected based on the performance and efficiency requirements of the specific application scenario.

B.3.3 Performance under Different Sampling Parameters

The performance of ICTL on different cooling rates and minimum temperatures is shown in Table 8. It can be observed that the performance of ICTL generally improves as the cooling rate increases and the minimum temperature decreases. However, after reaching a threshold, the performance no longer shows significant changes. Overall, the impact of temperature on performance is

within a certain range. Thus, for scenarios with limited computational resources and where performance sensitivity is relatively low, a combination with fewer iterations can be directly used (e.g., minimum temperature is $1e-4$, cooling rate is 0.9). For applications where performance is more critical, the threshold of performance change can be identified by gradually reducing minimum temperature and increasing cooling rate, allowing the optimal performance point with minimal time cost to be found using the fewest iterations.

B.4 Experimental Dataset

Super-NI Super-NaturalInstructions (Super-NI) targets cross-task generalization under natural-language instructions across diverse NLP tasks (e.g., classification, extraction, rewriting). Following prior work, we use the English-task subset split commonly adopted in recent studies: 756 training tasks and 116 test tasks, with the test tasks grouped into six categories for analysis.

GSM8K GSM8K focuses on grade-school math word-problem solving that requires multi-step arithmetic reasoning; it contains 7,473 training problems and 1,319 test problems.

MATH MATH evaluates competition-level mathematical problem-solving with step-by-step solutions across multiple subjects; it provides 7,500 training problems and 5,000 test problems.

MetaTool MetaTool benchmarks tool-awareness and tool-selection for LLM agents; its ToolE dataset comprises about 21,127 user queries that trigger single- and multi-tool use. The benchmark constructs evaluation test sets for awareness and selection tasks, rather than providing a standard train/test split.

NesTools NesTools evaluates nested tool learning (multi-tool calls where outputs feed subsequent tools) and offers a carefully curated evaluation set of 1,000 instances (with extensive nested-call coverage) for testing; the benchmark does not define a separate training split.

B.5 Implementation Detail

During sampling, we first select 16 source tasks that are most similar to each target task. For each target task, we sample 128 demonstrations from the source tasks to be transferred. Since Super-NI labels more than one answer for some questions, we transfer each answer with the question separately. For the transferred results, we sample 512 demonstrations for the inference. We employ the 3-shot inference, selecting demonstrations for each test question based on the BM-25 similarity. The reason for the parameter selection in this part is discussed in §3.4.

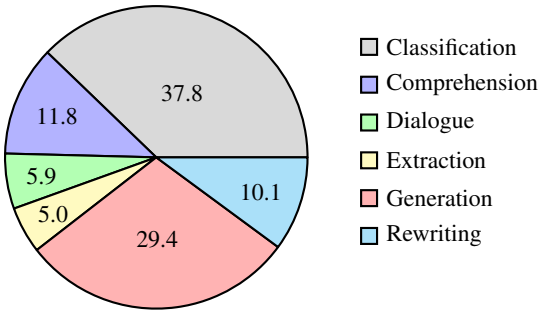


Figure 4: Category distribution of the Super-NI test set.

B.6 Category of Super-NI Test Tasks

The category of the Super-NI test set is shown in Table 9, where we follow the category of (Wang et al., 2024c). To better observe the impact of demonstration volume on transfer performance, we also count the distribution of demonstrations corresponding to different categories of tasks in the Super-NI test set, as shown in Figure 4.

C Additional Discussion

C.1 Efficiency Analysis of ICTL

C.1.1 Efficiency of Demonstration Synthesis

In this section, we provide a detailed analysis of the computational efficiency of ICTL. Our goal is to analyze how the efficiency of source sampling and target transfer impacts the overall runtime and resource utilization, particularly in terms of the source demonstration scale and model inference time.

Let N_s represent the total scale of the source demonstrations, N_s^S the scale of the sampled source demonstrations, and N_t^S the scale of the sampled target demonstrations. The symbol c_θ denotes the time taken by the sampling algorithm to process one single data point with parameter θ. Similarly, $c_\mathcal{M}$ represents the time for the model $\mathcal{M}$ to process a single data point.

$$c_\theta N_s N_s^S + c_\mathcal{M} N_s^S + c_\mathcal{M} N_s^S + c_\theta N_s^S N_t^S \quad (7)$$

Then, we can represent the total computational cost with Equation 7. In Equation 7, the first term

Category	Task ID
Classification	20, 50, 190, 199, 200, 201, 202, 226, 232, 233, 242, 290, 349, 391, 392, 393, 520, 614, 623, 640, **641**, 642, 738, 827, 828, 890, 935, 936, 937, 970, 1344, 1385, 1386, 1387, 1388, 1393, 1439, 1442, 1516, **1529**, 1554, 1612, 1615, 1624, 1640
Comprehension	**33**, 133, 249, 304, 329, 330, 401, **648**, 891, 892, 893, 1390, 1391, 1664
Dialogue	362, 879, 880, 1394, 1531, **1533**, **1534**
Extraction	36, **39**, **281**, 613, 620, 645
Generation	102, 219, 220, 288, 418, 500, 510, 569, 602, 619, 677, 743, 760, 769, 957, 1152, **1153**, 1154, 1155, 1156, 1157, 1158, 1159, 1161, 1342, 1356, **1358**, 1407, 1409, 1540, 1586, 1598, 1631, 1659, 1728
Rewriting	**34**, 35, 121, 402, 442, **670**, 671, 1195, 1345, 1557, 1562, 1622

Table 9: Category of the Super-NI test set. The tasks used for `GPT-4o` experiments are marked in **bold**.

represents the efficiency of source sampling, the second term corresponds to the target transfer, the third term describes the transfer verification, and the fourth term reflects the efficiency of the sampling of the synthesized demonstrations.

$$(c_\theta N_s + 2c_{\mathcal{M}})N_s^S + c_\theta N_s^S N_t^S \tag{8}$$

Based on Equation 7, we can derive Equation 8. From the equation, it can be observed that the total runtime is primarily dependent on N_s^S, which is the scale of the sampled demonstrations. Therefore, when computational resources are limited and the overall scale of the source demonstrations N_s is large or the model inference time $c_{\mathcal{M}}$ is high, we can reduce N_s^S to improve efficiency.

C.1.2 Efficiency of Inference

To evaluate the efficiency of ICTL during inference, we calculate the average input token numbers under different settings, as shown in Table 10. From the table, we can see that, during inference, the average token number of our method is similar to Direct and Synthesis. This is because the demonstration generation is offline, where during the inference, we only need to sample question-related demonstrations from the generation results, having a similar efficiency to the general ICL methods.

C.2 Synthesis Case Study

In this section, we conduct a case study on the data transferred by ICTL to gain a deeper understanding of how task transfer is performed. We investigate from two perspectives: capability transfer (Table 11, Table 13) and domain transfer (Table 12, Table 14). From these cases, we can observe that: *(i)* Capability transfer generally occurs when the source and target tasks are highly similar, where the definition or format of the source and target tasks are similar, our method can effectively understand the meaning of the source task and apply it to the target task; *(ii)* Domain transfer occurs when there is a significant difference between the source and target tasks, where the model leverages the original input information from the source task, which includes domain knowledge, while the answers or other information for the target task are generated independently by the model.

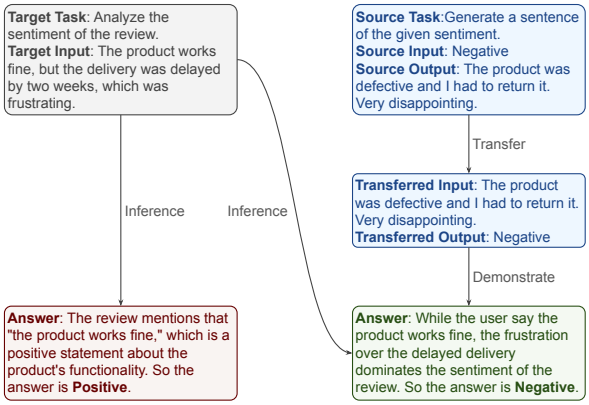


Figure 5: A case of Super-NI without (left) and with (right) ICTL. The correct answer is marked in green and the incorrect answer is marked in red.

C.3 Bad Case Study

To better understand how ICTL enhances reasoning performance, we analyzed a bad case of Super-NI, as shown in Figure 5. The figure illustrates that without ICTL, the model erroneously focuses on the phrase "`worked fine`", leading to an incorrect answer. However, with ICTL, the model is guided to more comprehensively evaluate the user input, thereby producing the correct result.

Setting	Zero	Direct	Single	Synthesis	ICTL
Average Tokens	95.7	257.3	156.7	278.7	262.3

Table 10: The average input token number during inference under different settings on Super-NI.

Source Data	**Definition**	**QA ZRE Question Generation on Subject Relation:** You will be given a context, a subject and a relation. Your task is to generate a question based on the subject and relation. The generated question should include the given subject. Try to use a minimum number of words that are not present in either context, subject or relation while generating question.
	Input	Context : Blind Company was shot in Bicheno, Tasmania in September 2008. Subject : Blind Company Relation : narrative location
	Output	Which place is Blind Company in?
Transferred Data	**Definition**	**Scitail1.1 Classification:** You are given two sentences. You have to find if there is entailment or agreement of the Hypothesis by the Premise. From the given pair of sentences, you should identify if there is enough information in the Premise to support the claim made in the Hypothesis. The Premise may not exactly be the same as Hypothesis. Your task is to return 'entails' if the premise supports hypothesis else return 'neutral'.
	Input	Premise: Blind Company was shot in Bicheno, Tasmania in September 2008. Hypothesis: Blind Company is in Bicheno.
	Output	entails

Table 11: The case study of the capability transfer for the classification task.

D Additional Experiment

D.1 Comparison with Different Filter Methods

To verify the validity of our proposed Equation 3 as a selection of similar examples, we compare the demonstration transfer performance using different source task sampling methods: BM25 (Robertson and Zaragoza, 2009), Contriever (Lei et al., 2023), and Dr.ICL (Luo et al., 2023). The experimental results are shown in Table 15, from which we can see that: *(i)* the sampling method of ICTL is better than other sampling methods, proving the effectiveness of ICTL; *(ii)* compared to Direct, the retrieval method of ICTL leads to a more significant performance improvement, indicating that our method is more sensitive to the quality of source demonstrations, where even a slight increase in the source quality can result in a substantial enhancement in final inference performance.

D.2 Target Sampling Divergence

To validate the effectiveness of Equation 2 as a sampling metric, we randomly sample 32 different sets of synthesized demonstrations. For each set, 128 demonstrations are randomly selected for each task, where the corresponding Equation 2 values and performance are shown in Figure 6. From the figure, we can observe the following: *(i)* As the Equation 2 value increases, the model performance shows a declining trend, indicating that the equation we proposed can effectively evaluate the divergence between the source demonstrations, the target task definition, and the synthesized demon-

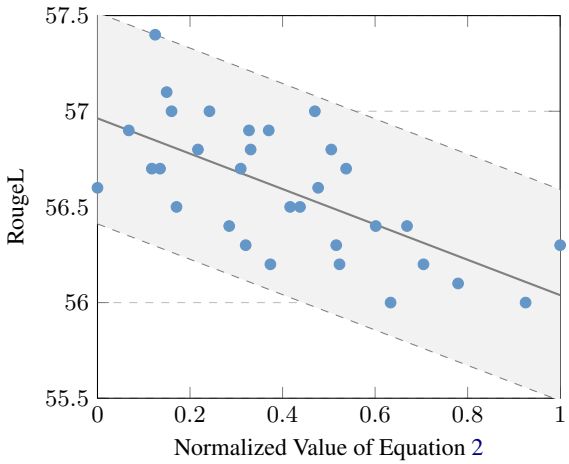


Figure 6: RougeL on the Super-NI test set using the 32 different sets of randomly sampled transferred demonstrations with different values of Equation 2 using Llama3.1-8b. To better observe the changes, we normalize the values of the X-axis.

Source Data	**Definition**	**XLWIC True or False Answer Generation:** In this task, you are given a word, followed by two sentences. Your task is to figure out whether both the sentences use the aforementioned word with the same meaning. You should respond with 'True' if the words in both sentences share the same meaning, and 'False' otherwise.
	Input	spring Sentence1: I spent my spring holidays in Morocco. Sentence2: He will hold office until the spring of next year.
	Output	False
Transferred Data	**Definition**	**ANLI R2 Entailment:** In this task, you will be presented with a premise and a hypothesis sentence. Determine whether the hypothesis sentence entails (implies), contradicts (opposes), or is neutral with respect to the given premise. Please answer with "Contradiction", "Neutral", or "Entailment".
	Input	Premise: The spring season is a time of renewal and growth, often associated with warmer weather and longer days. Hypothesis: He will hold office until the spring of next year.
	Output	Neutral

Table 12: The case study of the domain transfer for the classification task.

Source Data	**Definition**	**Para-NMT Paraphrasing:** This is a paraphrasing task. In this task, you're given a sentence and your task is to generate another sentence which express same meaning as the input using different words.
	Input	someone other than the owner must have known it .
	Output	someone , outside the owner , must have known about that .
Transferred Data	**Definition**	**Ollie Sentence Answer Generation:** Given two noun phrases (arguments) and relationship between them, form a sentence that expresses these arguments with the given relationship.
	Input	Relationship: 'known' Argument/Subject 1: 'someone other than the owner' Argument/Subject 2: 'it'
	Output	someone other than the owner must have known it.

Table 13: The case study of the capability transfer for the generation task.

strations, which in turn helps assess model performance; *(ii)* The variation in all experimental results is less than two points, suggesting that sampling synthesized demonstrations has a relatively small impact on performance, matching the results in Table 3, since we also select the question-related demonstrations during subsequent inference, which overlaps the effectiveness of the target sampling.

D.3 The Transfer between Source Demonstrations and Target Demonstrations

To investigate the relationships between different source tasks and target tasks, we static the number of target demonstrations corresponding to various source tasks after source sampling. The statistical results are shown in Figure 7, from which we can see that: *(i)* Almost all source tasks contribute to various target tasks, showing the necessity of sampling source demonstrations from different source tasks; *(ii)* Among all source tasks, *Generation* provides significant assistance to all categories of target tasks, indicating that the generalization capability of different source tasks to various target tasks differs; *(iii)* All categories of source tasks contribute demonstrations to target tasks for transfer learning, suggesting that when choosing source tasks to be sampled, it is essential to cover different categories of tasks to help target tasks acquire diverse capabilities and domain knowledge.

D.4 Pass Rate of Transfer Verification

To verify the quality of the transfer results across different target tasks, we report the pass rates of transfer verification across various task categories, as shown in Table 16. From the table, we can

Source Data	**Definition**	**Peixian Rtgender Sentiment Analysis:** Given a 'poster' sentence and a corresponding 'response' (often, from Facebook or Reddit)classify the sentiment of the given response into four categories: 1) Positive, 2) Negative, 3) Neutral, and 4) Mixed if it contains both positive and negative.
	Input	Poster: La edad hace de las suyas con mis ojitos. Aging is getting to my eyes. OMG!!!! Responser: sorryy jeje eso dije
	Output	Neutral
Transferred Data	**Definition**	**Reddit Tifu Title Summarization:** In this task, you are given a Reddit post as a text. Your task is to generate a title for this text. The title should start with TIFU by, followed by a situation that caused humor. The title should contain 7-12 words, ideally.
	Input	Text: La edad hace de las suyas con mis ojitos. Aging is getting to my eyes. OMG!!!!
	Output	TIFU by letting aging ruin my eyes in seconds

Table 14: The case study of the domain transfer for the generation task.

Retriever	Direct	ICTL
BM25	46.2	55.8
Contriever	46.5	56.3
Dr.ICL	48.4	58.7
ICTL	**48.8**	**60.3**

Table 15: The RougeL of ICTL filtering source task data with different retriever under two settings (Direct, ICTL) on Super-NI using Llama3.1-8b. The best performance is marked in **bold**.

Category	Pass Rate (%)
Classification	85.6
Comprehension	68.4
Dialogue	76.5
Extraction	80.8
Generation	61.3
Rewriting	66.6
Overall	74.1

Table 16: The pass rate of the transfer verification of ICTL on the Super-NI test set using Llama3.1-8b.

observe that: *(i)* For all task categories, the synthesized demonstrations of ICTL achieve a pass rate of over 60%, indicating that the synthesized results generally satisfy the requirements of the target tasks; *(ii)* Compared to tasks with more definite answers (e.g., Classification, Extraction), tasks with more open-ended answers (e.g., Generation, Rewriting) exhibit lower pass rates, since during transfer for these tasks, the model struggles to determine the appropriate answer format based on the task definition, leading to poorer transfer results.

Metric	Single	+ ICTL	Multiple	+ ICTL
EM	39.7	44.0	41.5	45.6
RougeL	54.7	60.3	57.6	60.4

Table 17: The performance of ICTL with and without additional human labeling using Llama3.1-8b. **Single** denotes only using the example of each target task. **Multiple** denotes using additional human-labeled demonstrations provided by Super-NI.

D.5 Combine ICTL with Human-Labeling Demonstrations

To verify the performance of our method in the presence of human-labeled demonstrations, we conduct experiments using additional demonstrations labeled by humans. For each test task, we utilize the dataset excluding the 100 test instances as the demonstration pool for the experiments. We perform two sets of experiments: one using only human-labeled demonstrations and the other combined with the demonstrations transferred by ICTL. The experimental results are shown in Table 17. From the table, we can see that compared to the results using only human-labeled demonstrations, our method achieves further performance improvements, demonstrating the effectiveness in augmenting demonstrations labeled by humans.

D.6 Performance of ICTL cross Different Domain

To evaluate the performance of ICTL across different domains for the same task, we conduct cross-domain experiments. Since all different tasks of Super-NI exhibit some variation, we opt to use

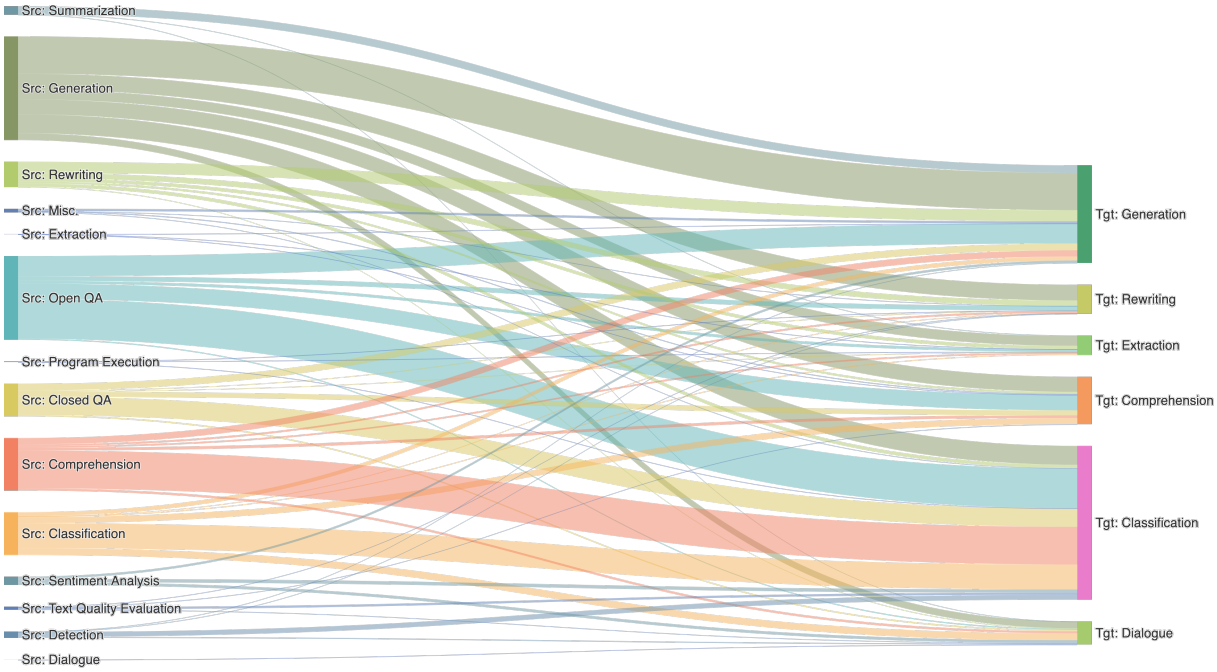


Figure 7: The Sankey figure of the transfer between different source categories and target categories. The category follows (Wang et al., 2024c).

Category	Zero	Direct	Single	Synthesis	Ours
Name Entity Recognition	28.2	84.4	85.0	84.6	**85.4**
Natural Language Inference	21.1	21.7	21.0	22.5	**24.8**
Question Answering	60.6	62.5	64.2	62.3	**64.8**
Sentiment Analysis	71.5	73.8	70.0	70.8	**74.0**
Overall (EM)	33.2	36.8	34.8	35.3	**39.9**
Overall (RougeL)	45.4	60.6	60.0	60.0	**62.2**

Table 18: The cross-domain performance of ICTL on BOSS (Yuan et al., 2023) under different settings present in §3.1 using Llama3.1-8b. The performance of each category is evaluated with RougeL. We delete all toxic detection questions because the security restrictions of the model we use lead to refusal to answer questions with sensitive words. The best performance of each category is marked in **bold**.

BOSS (Yuan et al., 2023) for the experiments, which standardizes the input-output format for data across different domains within the same task, allowing for a more accurate evaluation of cross-domain performance. The experimental results are shown in Table 18, from which we can observe the following: *(i)* Under the setting of the same task across different domains, our method still yields performance improvements, demonstrating its effectiveness in cross-domain scenarios. *(ii)* Apart from our method, *Direct* achieves the best performance, since despite being in different domains, the task and input-output format are identical, allowing the model to learn how to perform accurate reasoning from demonstrations in other domains of the same task.

Definition	EM	RougeL
Auto-ICL	42.3	59.1
Human-Labeled	44.0	60.3

Table 19: The performance of ICTL using task definitions synthesized by LLMs and labeled by humans on Super-NI.

D.7 Performance of ICTL with Synthesized Definitions

Considering that humans could label no task definition in the real application, we discuss the performance of ICTL using the synthesized definitions in this section. We employ Auto-ICL (Yang et al., 2024) to synthesize task definition, where the experiment results are shown in Table 19. From the table, we can find that the performance degrada-

tion caused by synthetic definition is not significant. This is because the performance of our method is not particularly sensitive to the similarity between the source task and target task definitions, as shown in Figure 3c.