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 using LLMs. However, the quality of the demonstrations synthesized 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 that 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. The results on 8 datasets of 4 mainstream tasks show that ICTL has achieved 1.7% relative performance improvement compared with the existing methods, achieving the state-of-the-art (SOTA) performance of demonstration synthesis.

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., 2024b). 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 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 synthesis results of previous iterations.

However, the synthesis using LLMs 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., 2024c). 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., 2024b; 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 demonstrations that are highly similar to the target task. Then, taking the sampled results and the target task definition as input, we transfer the sampled demonstrations to the target task using LLMs.

To validate ICTL, we conduct experiments using five mainstream LLMs on eight datasets across four mainstream tasks. Experimental results demonstrate that ICTL achieves a relative performance improvement of 1.7% over existing baselines, which establishes a new state-of-the-art

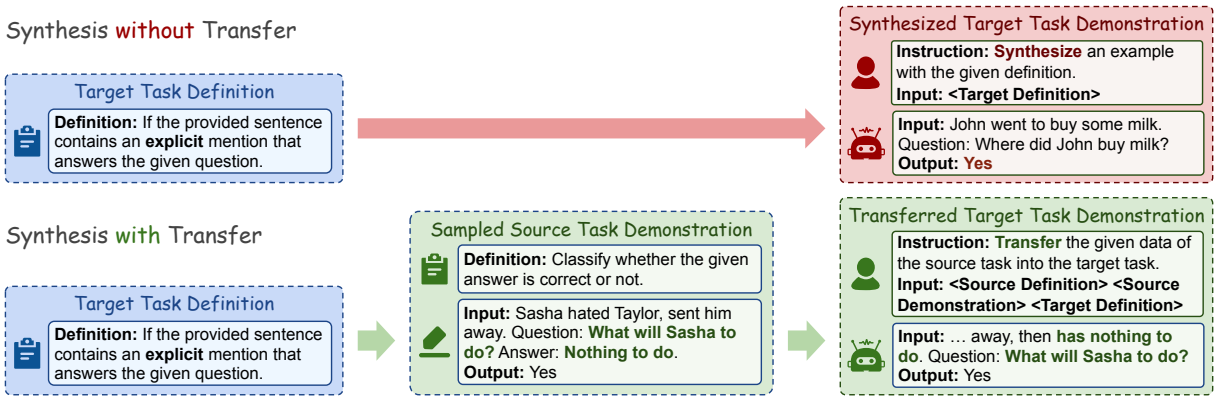

Figure 1: Comparison between previous demonstration synthesis methods (top) and our method (bottom). The blue part denotes the definition of the target task. The previous method synthesizes the demonstration from scratch, while the model misinterprets the definition and generates a demonstration with 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**.

(SOTA) among demonstration synthesis methods, verifying the effectiveness of ICTL. Further experiments demonstrate that, compared to single-task settings, leveraging demonstrations from multiple tasks enhances the performance of ICTL further, validating the effectiveness of the transfer learning employed in our method.

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;
- ICTL achieves 1.7% improvement compared with existing baselines, achieving new SOTA results on the demonstration synthesis.

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 §3.5 and 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 demonstrations. 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:

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

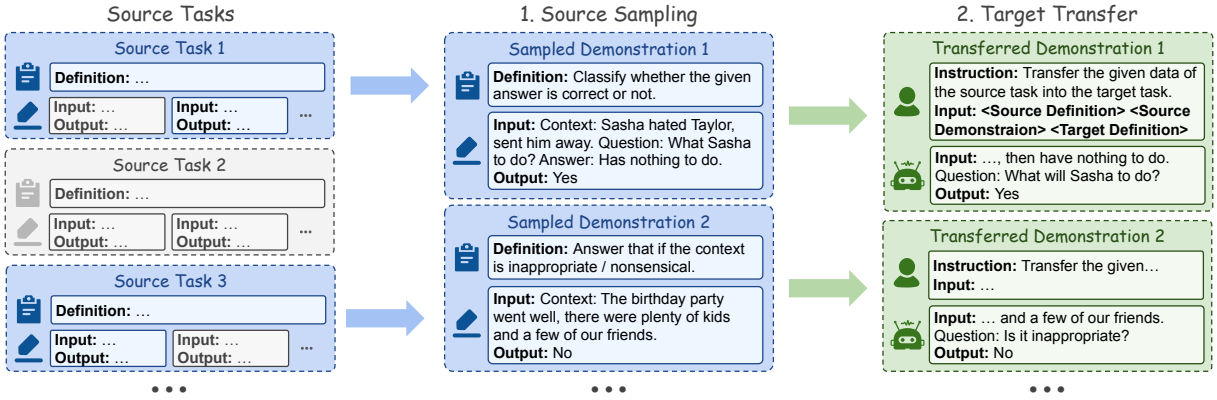

Figure 2: The illustration of ICTL, taking the target task definition "*If the provided sentence contains an explicit mention that answers the given question*" as an example. ICTL consists of two steps: *(i)* Source Sampling: sample demonstrations similar to the target task from the source tasks; *(ii)* 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.3.

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. (2024b), 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 of source tasks 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.

Sample The sample step aims to sample the transferred target demonstrations with Equation 2, ensuring that the sampled transferred 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 target task.

3 Experiments

3.1 Experiment Setup

Dataset To comprehensively evaluate the effectiveness of ICTL, we conduct experiments on eight datasets across four mainstream tasks, including: *(i)* Math: GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021); *(ii)* Reasoning: ARC-Challenge (Clark et al., 2018), MMLU-Pro (Wang et al., 2024c); *(iii)* Code: HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021); *(iv)* Tool: MetaTool (Huang et al., 2024), NesTools (Han et al., 2025). Detailed descriptions of these datasets are provided in Appendix B.4. Since each task contains two datasets, for each task, when evaluating one dataset, we utilize another dataset as the source for synthesis. We employ pass@1 as the evaluation metric for Code datasets and Exact Match (EM) for all other tasks. Furthermore, to validate the generalizability of ICTL across a broader range of tasks, we also conduct experiments on Super-NI dataset in Appendix D.1.

Model We evaluate ICTL on Llama-3.1-8B/70B-Instruct (Llama-3.1) (Meta, 2024), DeepSeek-R1-Distill-Llama-8B (Llama-R1) (DeepSeek-AI et al., 2025), Qwen3-8B (Yang et al., 2025), and `gpt-5-nano` (OpenAI, 2025). Given that these models cover different families, scales, and closed-source architectures, they comprehensively validate the effectiveness of ICTL.

Baseline To verify the effectiveness of ICTL, we compare it with existing demonstration synthesis methods, including DAIL (Li et al., 2023), DDG (Wang et al., 2025c), and SynAlign (Ren et al., 2025). To ensure a fair comparison with ICTL, when evaluating on one task, we provide the demonstrations of all datasets to the given task as the demonstration pool for these baselines.

Setup We use BGE-EN-ICL (Chen et al., 2023a) to embed task definitions and demonstrations for sampling. For each target task, we sample 128 demonstrations from the source tasks to be transferred. For the transferred results, we sample 256 demonstrations for the inference. We employ the 3-shot inference, selecting demonstrations for each test question based on the BM-25 similarity, as demonstration selection is not the main topic of this paper. Our experiment is on two A100-80G, and each setting takes about 2 hours to run. All implementations are based on Transformers (Wolf et al., 2020) and vLLM (Kwon et al., 2023).

3.2 Main Experiment

The main experimental results are presented in Table 1. As shown in the table, ICTL achieves a relative performance improvement of 1.7% compared to other baselines. This establishes a new SOTA for demonstration synthesis methods, demonstrating the effectiveness of our method. We present the results of ICTL transferring with cross-task demonstrations in §3.4, which achieve better performance. Furthermore, we observe the following conclusion from the table:

Model As the capability of the base model increases, the performance gains yielded by ICTL diminish. This observation aligns with findings in Wang et al. (2025b), as models become more capable, they can sufficiently understand and solve the given tasks on their own, reducing their reliance on guidance from demonstrations.

Model	Method	Math		Reason		Code		Tool	
		GSM8K	MATH	ARC-C	MMLU-Pro	HumanEval	MBPP	NesTools	MetaTool
Llama-3.1-8B	Zero	83.6	49.0	82.1	50.4	65.9	54.9	50.2	57.2
	DAIL	83.3	49.2	82.8	51.0	66.8	54.9	50.7	58.0
	DDG	84.5	49.4	82.5	50.6	66.9	56.1	49.8	57.2
	SynAlign	84.1	50.6	82.3	52.6	67.1	57.0	51.6	57.8
	ICTL	**87.2**	**51.2**	**85.2**	**53.7**	**69.7**	**58.4**	**52.2**	**59.9**
Llama-3.1-70B	Zero	83.7	63.4	85.6	54.9	75.4	65.7	55.0	82.5
	DAIL	84.5	63.0	87.2	55.8	76.4	65.5	56.8	84.8
	DDG	84.3	63.0	85.6	54.5	76.8	66.4	55.1	85.5
	SynAlign	84.8	65.8	87.8	57.1	77.2	67.7	56.4	87.9
	ICTL	**86.7**	**66.8**	**88.8**	**58.0**	**79.0**	**69.0**	**58.0**	**89.2**
Llama-R1-8B	Zero	83.8	70.0	88.0	58.7	84.8	82.2	59.2	70.4
	DAIL	85.3	70.6	89.9	59.5	86.1	83.2	59.0	71.2
	DDG	83.3	69.8	88.6	59.7	84.3	81.8	60.0	74.9
	SynAlign	85.4	71.6	88.5	61.1	87.2	82.7	59.8	71.7
	ICTL	**86.5**	**73.0**	**90.9**	**61.7**	**88.0**	**85.3**	**62.0**	**78.7**
Qwen3-8B	Zero	89.7	76.2	89.4	61.5	88.3	86.1	61.2	70.5
	DAIL	90.4	78.0	90.8	63.2	89.1	87.7	62.5	72.3
	DDG	89.2	77.6	90.9	61.6	89.7	87.3	60.8	70.7
	SynAlign	89.9	76.6	91.7	63.3	90.1	87.2	61.9	72.3
	ICTL	**92.2**	**79.0**	**92.4**	**64.4**	**91.3**	**88.8**	**63.8**	**73.6**
`gpt-5-nano`	Zero	91.4	85.0	92.3	66.8	92.5	90.4	64.0	84.4
	DAIL	92.4	86.8	92.3	66.1	93.4	90.5	64.1	84.1
	DDG	91.3	85.0	92.1	66.2	92.4	90.6	64.4	84.9
	SynAlign	92.9	86.0	92.4	66.5	93.1	90.9	64.3	86.0
	ICTL	**93.8**	**87.6**	**93.1**	**67.4**	**93.4**	**91.2**	**64.5**	**87.3**

Table 1: Performance of ICTL against other demonstration synthesis baselines across various datasets and models. Zero denotes the zero-shot setting. ARC-C denotes ARC-Challenge. Due to API costs, we evaluate `gpt-5-nano` on a random sample of 128 instances per dataset. The best performance in each setting is marked in **bold**.

Dataset Under the same task setting, compared with a more challenging dataset, ICTL achieves a more pronounced performance gain over Zero when applied to the other easier dataset ($+3.3\%$ vs. $+2.6\%$). This is because, for the same task, demonstrations from a harder dataset typically contain richer domain knowledge or more sophisticated reasoning patterns, which can better guide the solution of simpler problems. In contrast, since solving the easier dataset generally requires simpler reasoning, its guidance effect on more complex problems is relatively weaker. Notably, however, even transferring from an easier dataset to guide the solving of a harder dataset still brings performance improvements. This suggests that even simple data can contain knowledge or reasoning strategies that the model lacks, thereby enhancing its ability to solve more challenging datasets to some extent.

Task ICTL yields more significant performance gains on Tool datasets compared to other datasets. This is attributed to the fact that, according to model training reports (DeepSeek-AI et al., 2025; Yang et al., 2025), Math and Code data account for a larger proportion of the post-training data compared to Tool data. Furthermore, Tool datasets involve distinct tool-specific characteristics, necessitating demonstrations to guide the model in learning how to utilize different tools effectively. Consequently, ICTL delivers more substantial performance improvements on Tool tasks.

3.3 Ablation Study

To verify the effectiveness of each component in ICTL, we conduct ablation studies of our method. The experimental results are shown in Table 2. The experimental results show that ablating any component leads to performance degradation, demonstrating the effectiveness of each part to ICTL. Besides, based on the table, we can see that:

Transfer Directly using sampled source demonstrations without transferring leads to the most severe performance degradation to our method. This is because, even when dealing with the same task, data from different datasets can differ in format or domain. Such differences can mislead the inference, causing ICL performance to drop, and it could even fall below zero-shot performance. This indicates that adapting source-task demonstrations to the target task through transfer is necessary to ensure the performance of ICTL.

Method	GSM8K	MATH	ARC-C	MMLU-Pro	HumanEval	MBPP	NesTools	MetaTool	Total
ICTL	87.2	51.2	85.2	53.7	69.7	58.4	52.2	59.9	64.7
- Transfer	$84.4_{(-2.8)}$	$50.0_{(-1.2)}$	$82.0_{(-3.2)}$	$50.6_{(-3.1)}$	$66.8_{(-2.9)}$	$55.8_{(-2.6)}$	$48.7_{(-3.5)}$	$56.1_{(-3.8)}$	$61.8_{(-2.9)}$
- Source Sample	$85.1_{(-2.1)}$	$49.4_{(-1.8)}$	$82.8_{(-2.4)}$	$51.8_{(-1.9)}$	$67.5_{(-2.2)}$	$56.7_{(-1.7)}$	$50.2_{(-2.0)}$	$57.6_{(-2.3)}$	$62.7_{(-2.0)}$
- Target Sample	$86.4_{(-0.8)}$	$50.0_{(-1.2)}$	$84.3_{(-0.9)}$	$52.7_{(-1.0)}$	$68.4_{(-1.3)}$	$57.7_{(-0.7)}$	$51.5_{(-0.7)}$	$59.2_{(-0.7)}$	$63.8_{(-0.9)}$

Table 2: The ablation experiments using Llama-3.1-8B on all experimental datasets. **Total** denotes the average performance on all datasets. *(i) - Transfer*: directly use the sampled source demonstration without transferring; *(ii) - Source Sample*: sample source demonstrations randomly without Equation 3; *(iii) - Target Sample*: use all transferred demonstrations without target sampling.

Source \Target	Math	Reason	Code	Tool
Zero	64.6	66.2	60.4	50.4
Math	67.7	67.0	61.5	51.0
Reason	65.5	69.5	62.5	52.0
Code	66.0	68.0	64.1	53.0
Tool	66.5	68.5	63.0	53.8
All	68.0	70.2	64.5	54.5

Table 3: The performance of ICTL when using cross-task demonstrations with Llama-3.1-8B. We present the average performance on all datasets of each task. The row name denote the source dataset and the column name denote the target dataset. **Zero** denotes the performance of the zero-shot setting. **All** denotes synthesis with demonstrations from all other datasets.

Method	EM	T(s)
Zero	49.0	–
DAIL	49.2	0.771
DDG	49.4	0.815
SynAlign	50.6	0.693
ICTL	51.2	0.623

Table 4: The performance and time cost on MATH using Llama-3.1-8B cross different method. **T** denotes the average time required to synthesize one single demonstration of each method.

Source Sample Removing source sampling also causes a sharp performance drop of 2.0% on average. This is because, without source sampling, ICTL uses random sampling of source demonstrations, which leads to many dissimilar source demonstrations being sampled, thereby decreasing performance. This result proves the necessity of sampling the source demonstrations according to their similarity to the target task before the transfer.

Target Sample Removing target sampling has the least impact on performance, causing only a 0.9% 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 Cross-Task Transfer

In this section, we investigate the effectiveness of ICTL in cross-task transfer scenarios. We conduct experiments using Llama-3.1-8B, and the results are presented in Table 3. From the table, we can observe the following: *(i)* When performing ICL with demonstrations transferred from other tasks, the performance is consistently lower than when using demonstrations derived from the task's own data. This reflects the impact of inter-task similarity on synthesis quality, where a more similar source task leads to greater performance gains on the target task. *(ii)* The performance improvement achieved by the All setting surpasses that of any other individual task. This indicates that ICTL effectively selects source task data relevant to the target task from the combined pool, thereby obtaining higher-quality demonstrations through transfer compared to using a single source task. *(iii)* Across all settings, the performance of ICTL is superior to the zero-shot baseline, suggesting that even source task data with low relevance can provide a certain degree of guidance for the target task.

3.5 Efficiency Comparison Cross Different Methods

To better compare the efficiency of ICTL against existing demonstration synthesis methods, we compare the average time required to synthesize a single demonstration on the same device. The experimental results are shown in Table 4. From the table, we can observe that ICTL achieves better performance with a comparable time cost compared with existing demonstration synthesis methods, demonstrating the efficiency of ICTL.

3.6 Impact of Different Parameter

To better guide the practical application of ICTL, in this section and shad the light of the future research, in this part, we analyze the impact of different parameters in ICTL based on Llama-3.1-8B.

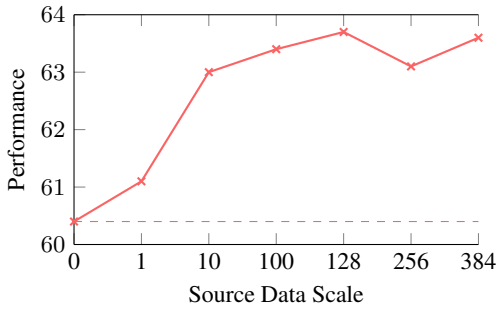

Figure 3: The performance change of ICTL with different source data scales using Llama-3.1-8B. The doted line denotes the zero-shot performance.

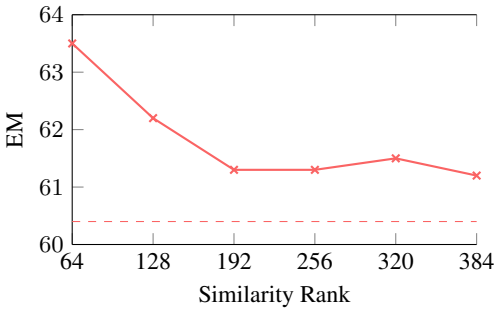

Figure 4: The performance change of ICTL with different source data similarity using Llama-3.1-8B. The x-axis represents transfer using source task data with similarity ranks in $[1, 64]$, $[65, 128]$, $[129, 192]$, ... from left to right, which implies that the similarity of the source task data gradually decreases from left to right. The doted line denotes the zero-shot performance.

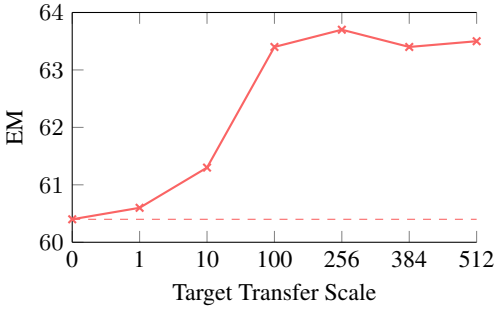

Figure 5: The performance change of ICTL with different target scales using Llama-3.1-8B. The doted line denotes the zero-shot performance.

3.6.1 Source 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 3. 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 acquire as many demonstrations as possible to ensure that there are enough different abilities and knowledge for the target task during ICL.

Notably, compared to not using transfer learning, even transferring with one source demonstration can effectively improve the performance of the target task. This is because: *(i)* Even by using a single source demonstration, we can synthesize numerous 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 their inherent knowledge, thereby enhancing the inference performance of ICL.

3.6.2 Source Similarity

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. The experimental results are shown in Figure 4, from which we can observe the following: *(i)* As similarity decreases, the performance of ICTL shows an overall downward trend, indicating the importance of the similarity between source task data and the target task when using ICTL. *(ii)* Even when using source data with very low similarity, the performance of ICTL remains higher than that of the zero-shot baseline, demonstrating the robustness of the low-quality source data.

3.6.3 Target Scale

Due to the limitations of computational resources in practical applications, the scale of the transferred demonstrations could be restricted. Therefore, we evaluate the performance of ICTL under different scales of transferred demonstrations, as shown in Figure 5. From the figure, we can observe the following: *(i)* Even if only 10 demonstrations are obtained through transfer, our method achieves better

performance than it does without transfer. As the scale of transferred demonstrations increases, the performance improves accordingly, demonstrating the necessity of sufficient transferring; *(ii)* 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 256 transferred demonstrations. Further increasing the scale does not yield new high-quality demonstrations, thereby lowering performance as a result of mixing in 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. (2025d) 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., 2024b; 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 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 to guide the source demonstration sampling, aiming to minimize transfer errors. 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. The experimental results show that ICTL achieves 1.7% relative improvement compared with previous synthesis methods, demonstrating new SOTA performance in the demonstration synthesis.

Limitations

We only adapt the experiment on the English datasets, where expanding to more languages can better reflect the effectiveness of ICTL.

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

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

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.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

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.

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. *Preprint*, arXiv:1803.05457.

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*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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)*.

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 *The Thirteenth International Conference on Learning Representations*.

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. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14620–14642, Vienna, Austria. Association for Computational Linguistics.

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

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.

Dawei Li, Yaxuan Li, Dheeraj Mekala, Shuyao Li, Yulin wang, Xueqi Wang, William Hogan, and Jingbo Shang. 2023. Dail: Data augmentation for in-context learning via self-paraphrase. *Preprint*, arXiv:2311.03319.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024a. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

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. 2024b. 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. 2024c. 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.

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. 2025. Introducing gpt-5. `https://openai.com/index/introducing-gpt-5/`.

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.

Jiyuan Ren, Zhaocheng Du, Zhihao Wen, Qinglin Jia, Sunhao Dai, Chuhan Wu, and Zhenhua Dong. 2025. Few-shot llm synthetic data with distribution matching. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 432–441, New York, NY, USA. Association for Computing Machinery.

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 *The Thirteenth International Conference on Learning Representations*.

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 M. Rotskoff. 2025. Features are fate: a theory of transfer learning in high-dimensional regression. In *Forty-second International Conference on Machine Learning*.

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.

Dingzriui Wang, Xuanliang Zhang, Keyan Xu, Qingfu Zhu, Wanxiang Che, and Yang Deng. 2025b. Learning-to-context slope: Evaluating in-context learning effectiveness beyond performance illusions. *Preprint*, arXiv:2506.23146.

Wuyuqing Wang, Erkun Yang, Zilan Zhou, and Cheng Deng. 2025c. 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. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Wuyuqing Wang, Erkun Yang, Zilan Zhou, and Cheng Deng. 2025d. 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. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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. 2024b. 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.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024c. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. 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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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*.

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 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 5: 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 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.

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.

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 Llama-3.1-8B. The best result is marked in **bold**.

B.3.2 Comparison with Other Methods

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

	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.

- 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

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. In this paper, we adapt experiments with MATH-500 (Luo et al., 2024a).

ARC-Challenge The ARC-Challenge (AI2 Reasoning Challenge) serves as a rigorous benchmark for question-answering systems, specifically targeting grade-school science questions that require complex reasoning and common sense, rather than simple retrieval. It consists of a subset of the larger ARC dataset, filtered to include only "hard" questions that existing algorithms initially failed to answer. The dataset is divided into a training set of 1,119 questions, a development set of 299 questions, and a test set of 1,172 questions, providing a focused environment for testing logical inference capabilities.

MMLU-Pro MMLU-Pro is an advanced evaluation suite designed to address the limitations of the original MMLU by increasing difficulty and reducing noise. It spans 14 diverse domains (such as biology, law, and psychology) and challenges models with multi-step reasoning tasks, often increasing the number of answer choices to ten to mitigate random guessing. Unlike datasets designed for fine-tuning, MMLU-Pro is primarily an evaluation benchmark; it consists of a massive test set of approximately 12,032 questions and a small validation set of 70 examples used for few-shot prompting or hyperparameter tuning.

HumanEval HumanEval is a widely used benchmark created by OpenAI to evaluate the functional correctness of code generation models. It consists of 164 hand-written programming problems (the test set) that are not present in standard training corpora, ensuring that the model must generate new code rather than memorizing existing solutions. Each problem includes a function signature, a docstring, and unit tests. As a zero-shot evaluation benchmark, HumanEval does not contain a training set, requiring models to synthesize Python code solely based on the provided natural language

specifications.

MBPP The Mostly Basic Python Programming (MBPP) dataset evaluates the ability of language models to solve entry-level programming concepts using Python. It features crowd-sourced problems that include short problem descriptions, code solutions, and automated test cases. The standard dataset comprises 974 total problems, typically split into a training set of 374 examples, a test set of 500 examples, a validation set of 90, and 10 examples for few-shot prompting. This structure allows researchers to evaluate both few-shot learning and fine-tuning performance on fundamental coding tasks.

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.

C Additional Discussion

C.1 Efficiency Analysis of ICTL

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.

$$c_\theta N_s N_s^S + c_\mathcal{M} N_s^S + c_\theta N_s^S N_t^S \tag{7}$$

Then, we can represent the total computational cost with Equation 7. In Equation 7, the first term represents the efficiency of source sampling, the second term corresponds to the target transfer, the third 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.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 9, Table 11) and domain transfer (Table 10, Table 12). 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.

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 6. 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.

D Additional Experiment

D.1 Additional Experiment on Super-NI

To further evaluate the cross-task performance of ICTL, we adapt the experiment on Super-NaturalQuestions (Super-NI) (Wang et al., 2022), which covers 1600+ different tasks. To thoroughly

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

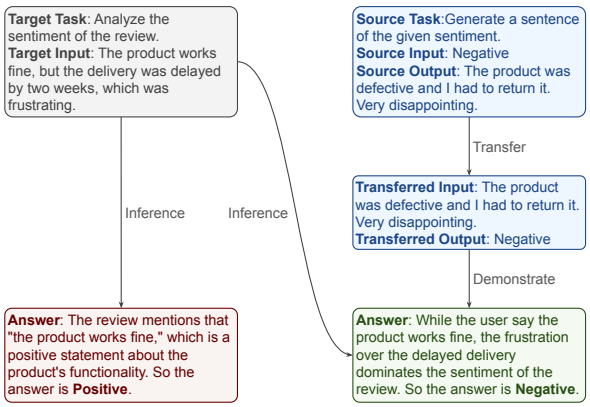

Figure 6: A case of Super-NI without (left) and with (right) ICTL. The correct answer is marked in green and the incorrect answer is marked in red.

analyze the effectiveness, we compare ICTL with the baselines on Super-NI:

- **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.

As the result shown in Table 13, 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.2. 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 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 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.

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 10: 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 11: The case study of the capability transfer for the generation task.

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.

Model Besides, compared to Llama-3.1-8B, 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.

D.2 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 14, 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.

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

Model	Category	Zero	Direct	Single	Synthesis	ICTL
Llama-3.1-8B	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 13: 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-2024-11-20`.

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 14: The RougeL of ICTL filtering source task data with different retriever under two settings (Direct, ICTL) on Super-NI using Llama-3.1-8B. The best performance is marked in **bold**.

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

Table 15: The performance of ICTL with and without additional human labeling using Llama-3.1-8B. **Single** denotes only using the example of each target task. **Multiple** denotes using additional human-labeled demonstrations provided by Super-NI.

D.3 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 7. 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 demonstrations, 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 2, since we also select the question-related demonstrations during subsequent inference, which overlaps the effectiveness of the target sampling.

D.4 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

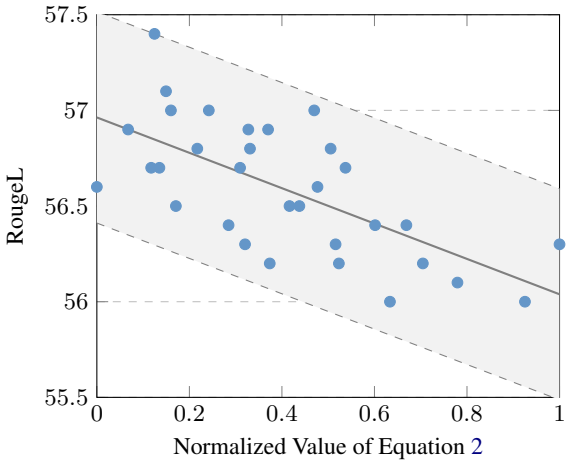

Figure 7: RougeL on the Super-NI test set using the 32 different sets of randomly sampled transferred demonstrations with different values of Equation 2 using Llama-3.1-8B. To better observe the changes, we normalize the values of the X-axis.

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

Table 16: The performance of ICTL using task definitions synthesized by LLMs and labeled by humans on Super-NI.

human-labeled demonstrations and the other combined with the demonstrations transferred by ICTL. The experimental results are shown in Table 15. 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.5 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 16. From the table, we can find that the performance degradation 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.