Logical DA: Enhancing Data Augmentation for Logical Reasoning via a Multi-Agent System

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs) have highlighted the importance of improving their reasoning capabilities. A critical challenge lies in the scarcity of high-quality reasoning data—characterized by diversity and rich supervisory signals—necessary for robust model training. While data augmentation (DA) methods have been leveraged to mitigate this scarcity, prevailing approaches often introduce noise and exhibit logical inconsistencies, thereby diminishing their utility for complex reasoning tasks. Moreover, existing DA paradigms predominantly isolate data synthesis from label validation, failing to unify these complementary processes within a cohesive architecture. To address these limitations, we introduce Logical DA, a multi-agent framework for enhancing reasoning-focused data augmentation in few-shot learning scenarios. Our system includes four agents operating through two synergistic phases: (1) diverse data generation, and (2) label verification. The system incorporates a reflection mechanism to continuously improve data quality by leveraging feedback from logical validation. We demonstrate the effectiveness of Logical DA through experiments on various tasks and datasets, achieving the highest average improvement in task accuracy in both fine-tuning and in-context learning paradigms, with an average improvement of 7.61% when applied to fine-tuning. Our code is available at https://anonymous.4open.science/r/acl25-E819

1 Introduction

Researchers have been focusing on the reasoning capability of large language models (LLMs) recently (OpenAI, 2024), where a key challenge is the limitation of high-quality reasoning data for model training. The ideal training data of reasoning is diverse and is rich in supervision information

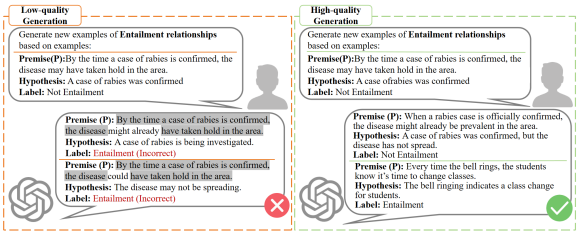

Figure 1: Low-quality data generation vs High-quality data generation

(Long et al., 2024), which requires costly manual annotations (Xu et al., 2025).

In view of this challenge, researchers have increasingly employed data augmentation (DA) techniques that enrich the dataset to alleviate the issue of data scarcity (Chen et al., 2023). Facilitated by the outstanding text generation capabilities of LLMs, researchers have employed LLMs for DA, including data generation derived from the original dataset and prompt-based label assignment (Ding et al., 2024), which have been experimentally verified effective (Dai et al., 2023a; Ding et al., 2023; Peng et al., 2024). Driven by the target tasks, such augmented data is then used for performance enhancement, via model training or integration into In-Context Learning (ICL) within the LLM (Wang et al., 2024a).

However, due to the inherent hallucination of LLMs and the lack of verification for generated labels, methods that rely solely on LLMs for data generation (Dai et al., 2023a) often produce noisy data, manifested as factual inaccuracies, label mismatches, or irrelevant content (Long et al., 2024), which sabotages the effectiveness of reasoning data augmentation, as shown in Figure 1. Moreover, the generated data often suffers from a lack of diversity that may jeopardize the model's generalization ability, which can be partially alleviated by controlled prompting (Yu et al., 2023a; Peng et al., 2024). Another challenge is that existing augmentation

techniques tend to focus on either data creation or data annotation, where a unified framework that effectively integrates both tasks is desired.

In this paper, we argue that an effective data augmentation framework, which aims to generate high-quality and logically sound data, requires two key capabilities: generating diverse samples and verifying logical correctness. Thus, inspired by the advantages of LLM-based Multi-Agent Systems in various domains (Chen et al., 2024), we propose a multi-agent system to enhance data augmentation in few-shot scenarios.

To jointly optimize data diversity and label consistency, we architect a multi-agent framework comprising four collaborative agents, structured into two main phases:

(1) **Diverse Data Generation**: In this stage, the Attribute Extraction Agent (AEA) performs an initial attribute extraction on the input data, generating a dynamic set of attributes specific to the data. The Filtering Agent (FA) then selects appropriate attribute combinations that are free from obvious logical conflicts and passes them to the Data Generation Agent (DGA) to generate preliminary transformed samples. Following this, FA filters out samples with richer variety, which are further processed by DGA to create high-diversity data.

(2) **Label Verification**: To mitigate label mismatches caused by hallucinations, we propose the Logical Label Verification Agent (LLVA), which grounds generated outputs in formal symbolic reasoning primitives. This improves the labeling accuracy of the generated reasoning data. Additionally, we implement a reflection mechanism based on logical validation feedback, allowing LLVA to feed label validation information back to DGA. This closed-loop architecture enables DGA to dynamically recalibrate its generation.

Our contributions are three-fold as follows:

- We first propose a complete multi-agent data augmentation framework, including the capabilities of generating data labeled with various categories and enhancing the fidelity of the generated data through logical validation.
- Our work is pioneering in focusing on the augmentation of logical reasoning data, alleviating the shortage of high-quality logical reasoning data and providing solid data support for enhancing the reasoning capabilities of LLMs, thus playing a foundational role.
- We conduct experiments on a wide spectrum of tasks and datasets, demonstrating the effectiveness of the augmented data by Logical DA in both fine-tuning and in-context learning

2 Related Work

Logical Reasoning. Early approaches to LLM-based reasoning operated directly on natural language representations (Wei et al., 2022; Clark et al., 2020; Zhou et al., 2022a). However, given that LLMs are essentially black-box probabilistic models, the reasoning outcomes derived in this manner could not be guaranteed to be faithful and reliable (Shanahan, 2024). Subsequently, a series of methods emerged that employed symbolic languages as the fundamental units of reasoning (Pan et al., 2023; Gao et al., 2023b; Xu et al., 2024b). Owing to their strict adherence to predefined reasoning rules and logic, these methods are deemed to be transparent and trustworthy.

Typically, LOGIC-LM (Pan et al., 2023) employs LLMs to convert natural language questions into symbolic representations, which are subsequently processed through symbolic executors for deterministic symbolic reasoning and result interpretation. Similarly, SymbCoT (Xu et al., 2024b) transforms natural language contexts into symbolic formats, generates stepwise procedural plans, and directs model reasoning along these structured guidelines. Building upon this approach, Aristotle (Xu et al., 2024a) implements an enhanced framework that guides LLMs in applying the resolution principle for logical deduction, achieving superior performance in logic reasoning tasks.

Differing from the aforementioned works that directly apply methods to enhance model reasoning capabilities, our objective is to employ symbolic reasoning to augment logical data. Additionally, we integrate the model's independent symbolic reasoning with the invocation of symbolic solvers, thereby more comprehensively ensuring the logical reasoning process.

Data Augmentation via LLMs. In recent years, data augmentation techniques based on LLMs have evolved from label-preserving to label-flipping approaches. Initially, methods directly utilized the demonstrations in prompts to guide models in generating new task-specific data (Kumar et al., 2020; Sahu et al., 2022; Dai et al., 2023b), which were subsequently employed for model training or in-context learning (Li et al., 2023; Su et al., 2024).

However, the data generated in this manner, lacking explicit generation criteria and patterns, may lead to spurious correlations.

In light of this limitation, several label-flipping data augmentation methods have emerged (Yoo et al., 2021; Yu et al., 2023a). These methods generate diverse data that are either related or contradictory by editing the attributes of the input data. FlipDA (Zhou et al., 2022b)introduces a data augmentation method that jointly uses a generative model (T5) and a classifier to generate label-flipped data. The key insight is that generating label-flipped data is more crucial for improving performance than generating label-preserved data. Inspired by this, CoTAM (Peng et al., 2024)instructed the LLM to flip labels to other labels in the dataset to get richer data.

In our work, we also employ a label-flipping approach similar to those mentioned previously. However, rather than confining labels to strictly similar or opposing categories, we strive to introduce a broader spectrum of labels (i.e. True, False and Unknown). This strategy ensures the diversity of the generated data.

LLM-Based Agents. The development of LLMs has demonstrated a notable capacity for interacting with environments and making decisions, thereby fulfilling expectations of intelligent agents (Li et al., 2024b). Drawing an analogy to real-world human society, where complex tasks are more efficiently, accurately, and systematically accomplished through the division of labor among multiple specialized departments, research on multi-agent systems based on LLMs has rapidly advanced across various domains. Examples include software development (Qian et al., 2024; Du et al., 2024), code generation (Islam et al., 2024), game simulation (Li et al., 2024a), and social network simulation (Gao et al., 2023a)).

The overarching strategy of replacing single-agent models with multi-agent models involves decomposing tasks into multiple distinct subtasks and assigning them to different agents. The relationships among these agents can be characterized as either linear pipelines (Yue et al., 2025; Liu et al., 2023; Shen et al., 2024), collective decision-making processes (Cheng et al., 2024; Liang et al., 2024), or iterative self-refinement (Wang et al., 2024b; Tang et al., 2024), ultimately converging to form a final decision.

In our proposed framework, following the linear workflow of three agents responsible for attribute extraction, data filtering, and data generation, we introduce a reflective mechanism for logical validation. This logical validation component provides feedback to the data generation agent, enabling it to verify and refine its output.

3 Preliminary

Problem Definition Our goal is to build a multi-agent system for data augmentation in logical reasoning tasks. This system aims to enhance the generation of high-quality and diverse reasoning data to improve the model's reasoning abilities. We define an LLM agent $a \in \mathcal{A}$ as a specialized model that takes text as input and returns text as output, specified by a unique identifier label l and a mapping $f : \mathcal{V} \rightarrow \mathcal{V}$. Each agent a_i is specialized in a specific subtask and is powered by an LLM $\mathcal{L}_i$.

For a text dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{k}$, where each sample consists of an input text x_i and a corresponding label y_i, data augmentation (DA) using large models involves utilizing prompts P to guide LLMs in generating novel augmented data from the given input text x_i as follows:

$$(x_i', y_i') = \mathcal{L}(P, (x_i, y_i)). \quad (1)$$

Logical Reasoning Data For a logical reasoning sample $(x_i, y_i) \in D$, the input text x_i typically consists of two text pairs (Pre_i, Hyp_i), where Pre_i is a set of premises $p_i, p_2, ...p_n$ with each p_i represents a logical statement, and Hyp_i is usually a statement that we aim to evaluate based on the premises. The label y_i for logical data refers to the logical relationship between the Pre_i and Hyp_i.

4 Method

4.1 Overview

In this paper, we propose Logical DA, a multi-agent system for reasoning data augmentation in few-shot scenarios. Our goal is to generate high-quality and high-diversity reasoning data with rich supervision signals (Long et al., 2024), which is essential for model training. We decompose the data augmentation task into two primary subtasks: 1) diverse data generation and 2) label verification. We design four agents to collaboratively complete the entire task. Specifically, our system employs a structured pipeline where agents specialize in specific tasks. To address the repetition issue in data generated by LLMs, we employ the Attribute Extraction Agent and the Filter Agent, which enhance the diversity

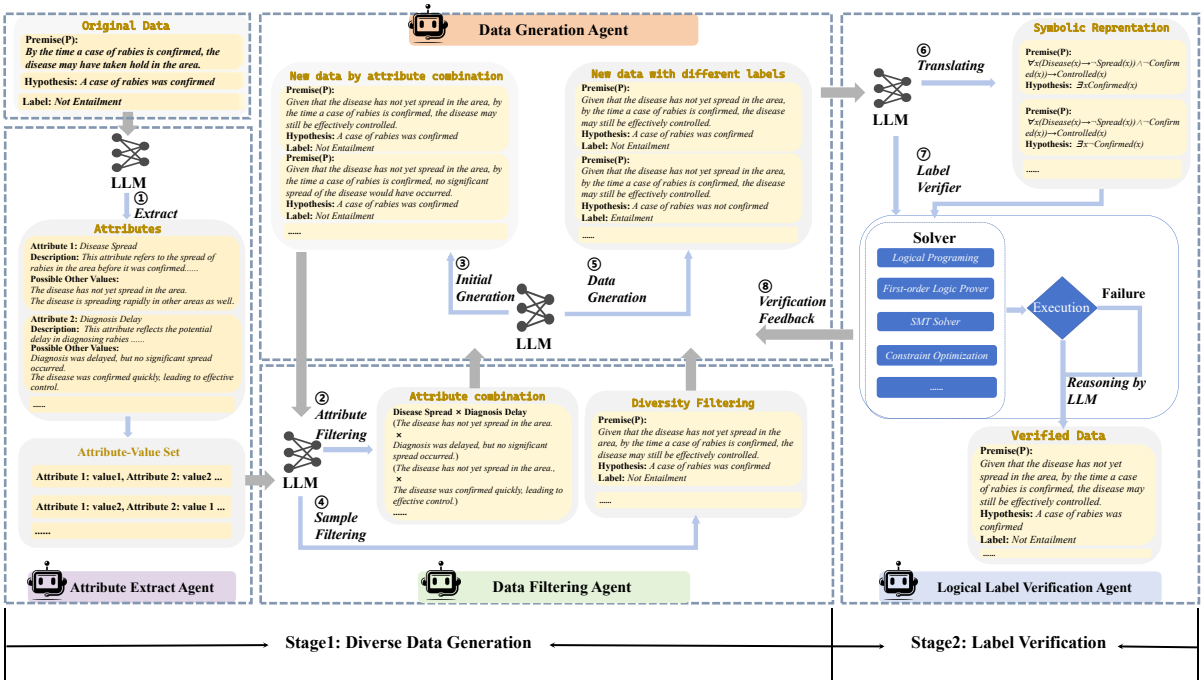

Figure 2: Overview of our multi-agent system, which contains two stages: using an existing BERT to divide the data according to difficulty and mixing up the samples to model's training from easy to hard. Best viewed in color.

of the data generated by the Data Generation Agent by introducing variations in the attributes of the original data (Sec.4.2). For label consistency, the Logic Verifier Agent ensures logical alignment between the generated data and its labels through symbolic reasoning (Sec.4.3). Moreover, Logical DA incorporates a reflection mechanism that iteratively refines the data based on external validation feedback, improving the overall quality of the generated reasoning data (Sec.4.3). The full framework is summarized in Figure 2 and demonstrated below.

4.2 The Stage of Diverse Data Generation

This is the stage more various samples are generated by extracting the attributes of the original data and varying these attributes to create new data. We design three agents to systematically accomplish this task, which will be introduced below in accordance with the pipeline.

Attribute Extract Agent. In our pipeline, the initial agent is the Attribute Extract Agent (AEA). Inspired by Yu et al. (2023b) and (Peng et al., 2024), the Attribute Extraction Agent (AEA) is developed to extract multiple attributes from the input sample. This process is akin to a latent space, where alternative possible values for each extracted attribute are identified, thereby forming a dynamic attribute-value set $\mathbf{Attr} = \{(attr_i, value_{i_1}, ..., value_{i_m})\}$.

Filtering Agent. The Filtering Agent (FA) primarily performs two operations:(1) filtering of attribute value combinations, and (2) data filtering based on diversity. First, FA employs the LLM to choose combinations of attribute values $\mathbf{Attr_{comb}}$ from the attribute-value set $\mathbf{Attr}$ provided by the previous agent, ensuring that the combinations selected are free from apparent conflicts. Based on the attribute combinations in $\mathbf{Attr_{comb}}$, the Data Generation Agent (DGA) generates the initial dataset X_{init}. Then, the FA selects the K most diverse samples from X_{init} to form the seed dataset X_{seed} for final diversified sample generation.

Specifically, for each $x \in X_{init}$, FA uses an embedding model $\mathcal{E}$, in this case, the sentence-BERT (Reimers and Gurevych, 2019) model—to map the data from the text space to the vector space, resulting in $\mathcal{E}(x)$. Then, we use cosine similarity to measure the diversity between samples. A smaller cosine similarity indicates greater diversity between samples. Consequently, FA selects the K samples with the smallest pairwise cosine similarity to form X_{seed} as Eq. 2.

$$X_{seed} = \underset{\{x_1,\ldots,x_K\}}{argmax} \sum_{1 \le i < j \le K} \left(1 - \cos\left(\theta\left(x_i, x_j\right)\right)\right) \quad (2)$$

Finally, FA passes the filtered data X_{seed} to the Data Generation Agent for the generation of diversified samples.

Data Generation Agent. The Data Generation

Agent (DGA) is responsible for the final stage of diversified data generation. The data generation methodology employed by the DGA is a two-step process: initially, it produces the preliminary transformed data X_{init} by means of attribute transformations; subsequently, for the X_{seed} refined by FA, it executes the conclusive multi-label data generation, adjusting the data to maintain alignment with predetermined label relationships.

In the first step of generation, DGA constructs a set of prompts P_{init} by combining the attribute-value pairs obtained from FA with the initial transformation prompt P_{init}. Then, DGA uses these prompts to invoke the LLM for transformation $\mathcal{L}(P_{init_{i'}}, Attr_{comb})$, producing the set X_{init}. This process is akin to latent space transformation: the original data is first mapped into an attribute space, and through attribute transformations in this space, it is then mapped back into the text space, resulting in diverse text samples.

In the second step of generation, in contrast to certain LLM-based data augmentation techniques (Dai et al., 2023a; Ding et al., 2023) that apply transformations while preserving labels, the DGA employs an LLM to generate data with label modifications since label modification augmentation provides valuable insights into the crucial components of a sentence that determine its label (Zhou et al., 2022b). Additionally, recent advancements (Peng et al., 2024) demonstrate that large language models have shown remarkable ability in controlling single-attribute transformations. Inspired by the aforementioned approaches, DGA employs a multi-label generation strategy for data generation, where the data is transformed into various logical relationships suitable for reasoning tasks. Practically, for a reasoning task dataset D with a label set C, the DGA generates data for a given data point $(x, y) \in D$ as Eq. 3

$$X_{new} = \{(x'_i, c_i) | x'_i = \mathcal{L}(x), c_i \in C\} \quad (3)$$

Where it is important to note that, in this context, we do not require x to be labeled data. Instead, we only need to know the task information associated with the data, which allows us to generate new data with various labels. The DGA then forwards the newly created data to the Logical Label Verification Agent for validation and amendment.

4.3 The Stage of Label Verification.

The task at this stage is to mitigate the incorrect labeling of generated data due to LLM hallucinations, which can adversely affect its utility. The integration of symbolic languages with large language models (Pan et al., 2023) has shown significant improvements in logical reasoning capabilities. To ensure better logical validation of generated reasoning data, we propose the Logical Label Verification Agent. This agent utilizes a specialized symbolic solver, combined with its own symbolic reasoning, to validate the generated labels.

Logical Label Verification Agent. The Logical Label Verification Agent (LLVA) validates the labels of the data generated by the Data Generation Agent (DGA), ensuring that the logical relationships between the generated data and their labels are consistent. In cases where discrepancies are found, the LLVA corrects the labels of the erroneous data, producing a validated dataset.

LLVA performs three primary actions: translation, solving, and label verification. In particular, the LLVA first translates a given input text into its corresponding symbolic representation. Then, depending on the inference task, it selects an appropriate symbolic solver to process the translated symbolic language. However, since the solver cannot fully execute the transformed symbolic representation, inspired by the enhanced reasoning capabilities of LLMs using symbolic language (Xu et al., 2024b), we leverage an LLM to perform reasoning based on symbolic operation rules. This enables logical verification of the generated data labels and allows for necessary corrections. Finally, the results from the logical reasoning are validated against the data's labels, and any inconsistencies are corrected. The validated data is then ready for downstream applications.

4.4 Agent reflection mechanism based on label validation feedback

To enhance the Data Generation Agent's (DGA) ability to generate reasoning data that better aligns with logical relationships, and inspired by studies (Gou et al., 2023) showing the effectiveness of external validation in improving agent capabilities, we have designed a feedback-based reflection mechanism for our multi-agent system. In this mechanism, information from the Logical Label Verification Agent (LLVA) during the label validation phase is fed back to the DGA to optimize its subsequent data generation.

Method	CB	RTE	FOLIO	ProofWriter	AR-LSAT	LogDed	Score	
	Acc.	*Acc.*	*Acc.*	*Acc.*	*Acc.*	*Acc.*	*Avg.*	*Δ (↑)*
Performance of Different LLM DA Methods								
BERT-base	55.94	52.21	34.64	39.44	22.79	18.33	37.22	–
-w/ AugGPT	72.01	54.62	36.76	36.44	23.23	17.55	40.10	+2.88
-w/ FlipDA++	57.14	56.55	37.08	49.05	23.51	22.22	40.92	+3.70
-w/ AttrPrompt	72.01	53.42	36.51	46.16	22.83	19.33	41.68	+4.46
-w/ COTDA	71.42	52.82	37.57	42.55	23.08	17.55	40.83	+3.61
-w/ COTAM	57.14	**56.55**	34.31	37.05	23.80	21.99	38.47	+1.25
-w/ SelfLLMDA	72.01	52.94	36.27	40.38	22.22	18.33	40.35	+3.13
-w/ Logical DA (Ours)	**74.10**	56.19	**39.14**	**51.85**	**24.38**	**23.33**	**44.83**	**+7.61**

Table 1: Comparison between our and the vanilla method applied to DA methods on the benchmarks. "Δ" denotes the improvement of our methods compared to the baselines. **Bold** denotes the best performance.

Model	FOLIO	ProofWriter	LogDed	*Avg.*
Llama-3.2-3b-instruct				
-w/ CoTDA	32.83	26.83	20.33	26.66
-w/ CoTAM	31.37	32.66	18.66	27.56
-w/ Logical DA (Ours)	**34.80**	**33.05**	**21.33**	**29.72**
GPT-3.5-turbo				
-w/ CoTDA	46.07	59.67	25.33	43.69
-w/ CoTAM	38.73	59.67	27.66	42.02
-w/ Logical DA (Ours)	**49.16**	**62.26**	**30.16**	**47.19**

Table 2: Experimental results of different data augmentation in ICL experiment. All values are average accuracy (%) of three runs with different seeds. Models are given 3 labeled data.

5 Experiments

5.1 Datasets and metrics

To investigate whether our method can effectively enhance model performance on reasoning tasks, we conduct extensive experiments on various reasoning tasks, including various tasks from FewGLUE (Schick and Schütze, 2020) and other complex reasoning benchmarks. *i.e.*, natural language inference (RTE, CB), deductive logical reasoning (ProofWriter (Tafjord et al., 2020), LogicalDeduction (LogDed) (Srivastava et al., 2022)), and logical reasoning tasks (AR-LSAT (Zhong et al., 2022b), FOLIO (Han et al., 2024)). To simulate the low-resource scenarios, we randomly select 10 samples per class from the training set for each task and use them for training the models. More information about the dataset can be found in Appendix A.

We use accuracy and EM (Exact Match) to evaluate the performance of the pre-trained models and LLMs on these datasets, respectively.

5.2 Compared Methods

We compared our method with other cutting-edge counterparts, including FlipDA (Zhou et al., 2022b), AugGPT (Dai et al., 2023a), AttrPrompt (Yu et al., 2023b), CoT Attribute Manipulation (CoTAM) (Peng et al., 2024), CoT Data Augmentation (CoTDA) (Peng et al., 2024) and Self-LLMDA (Li et al., 2024c). As the original FlipDA requires a large supervised dataset that is inapplicable to few-shot learning, we use an LLM-based variant FlipDA++ (Peng et al., 2024). More information about the baseline methods can be found in Appendix B.

5.3 Experiments settings

To comprehensively evaluate whether the data generated by our Logical DA can effectively enhance the model's logical reasoning capabilities, we conduct two sets of experiments: Fine-tuning and in-Context Learning (ICL). These experiments assess the performance of our method in both model training and large model context learning scenarios

1) **Fine-tuning experiment**: using the generated data to fine-tune the pre-trained models such as BERT (Devlin, 2018).

Specifically, we use the representative BERT-BASE model as the backbone PLM, and fine-tune them in a two-stage manner. Specifically, following many previous mixup methods, we first train the backbone PLMs (without using data augmentation) with a learning rate of 5e-5, and then continue fine-tuning the models using the data augmentation strategy with a learning rate of 1e-5. Note that our methods are only adopted in the second stage.

We set a maximum sequence length of 128 and a

batch size of 32. AdamW optimizer with a weight decay of 1e-4 is used to optimize the model. We use a linear scheduler with a warmup for 10% of the total training step.

2) **In-context learning (ICL) experiment**: extracting K samples from the generated data as examples and adding them into prompts to guide the generation of large language models such as Llama and GPT. We use GPT-3.5-turbo and Llama-3.2-3b-instruct as the base model, setting the parameter $K = 3$ and the temperature to 0.

Method	FOLIO	ProofWriter	LogDed	*Avg.*	Δ
Logical DA	39.04	51.85	24.38	38.42	–
-w/o Verifier	37.57	49.81	22.83	36.73	-1.69
-w/o Attr Trans	38.23	49.16	23.51	36.96	-1.45

Table 3: Experimental results of ablation study. All values are average accuracy (%) of three runs with different seeds.

5.4 Main results

The full results of fine-tuning and ICL are shown in Table 1 and Table 2, and we can find that:

Logical DA surpasses the cutting-edge counterparts in most settings. Our Logical DA brings much better performance improvements than the other counterparts, *i.e.*, up to +7.61% average score. Additionally, compared to the other LLM-based DA methods, Logical DA can also achieve superior performance. These results show the effectiveness of our Logical DA method.

Logical DA brings consistent and significant performance gains among all fine-tuning and in-context learning. Here, we verify whether our Logical DA can still work in the LLM scenarios. Taking some tasks as examples, we show the contrastive results in Table 2. It can be seen that, with the assistance of our Logical DA, LLM achieves much better performance against the baselines.

Logical DA works well in both model sizes. As shown in Table 2, Logical DA consistently outperforms other methods in terms of the average score for both Llama-3.2-3b-instruct and GPT-3.5-turbo models. Specifically, Logical DA achieves the highest average score of 29.72% for Llama-3.2-3b-instruct and 47.19% for GPT-3.5-turbo, indicating its effectiveness in enhancing model performance regardless of the model size.

5.5 Ablation Studies

We evaluate the impact of each component of our Logical DA, including *i*) diversity data generation, *ii*) label verification. By removing the corresponding agents from our multi-agent system and observing the resulting data generation, we assess their individual contributions.

Impact of diversity data generation. As mentioned in Sec 4.2, to obtain more diverse samples, we perform attribute transformations to generate the initial samples. To validate the importance of this step, we conducted a comparison experiment by removing the corresponding agents from the original system. As shown in Table 3, the performance dropped significantly, highlighting the critical role of this step.

Impact of Label Verification A key component of our method is the label verification stage. To evaluate its effectiveness, we removed the Logical Label Verification Agent (LLVA) from our system and compared the performance with that of the full Logical DA system. As shown in Table 3, when this agent is removed, the model's performance drops significantly, indicating that the LLVA agent plays an essential role in ensuring the logical consistency of the generated reasoning data, thereby enhancing the model's reasoning capabilities.

5.6 Direct Evaluation of Data Quality

The results of the above experiments demonstrate the exceptional performance of the data generated by our Logical DA for model training. In this section, we directly evaluate the quality of the generated data, including assessments of data diversity and data faithfulness (Long et al., 2024).

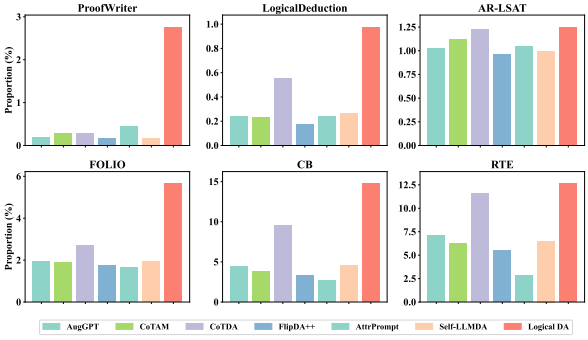

Figure 3: The proportion of unique words.

Divisity Evaluation Inspired by AttrPrompt (Yu et al., 2023a), we evaluate the diversity of the data generated by our system from two aspects: 1) the

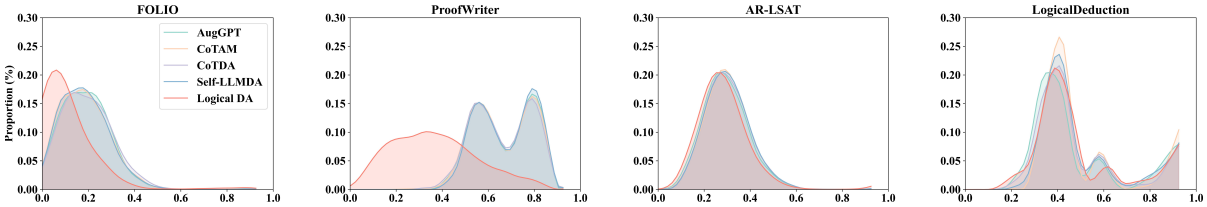

Figure 4: The distribution of data similarity.

proportion of unique words in the generated data and 2) the distribution of data similarity.

Figure 3 shows the proportion of unique words in all generated words, a metric often used to assess the diversity of the text. The results demonstrate that Logical DA generated a higher data diversity than any other baseline method.

Moreover, to gain a more intuitive understanding of the data distribution, we calculated the cosine similarity matrix among the data, normalized it, and plotted the distribution curve of the cosine similarity. As shown in Figure 4, Logical DA exhibits a more favorable similarity distribution compared to other methods. Specifically, the data generated by Logical DA shows a lower cosine similarity, indicating greater diversity and less redundancy.

Method	FOLIO	ProofWriter	LogDed	*Avg.*
FlipDA++	45.00	30.00	48.00	41.00
CoTDA	60.00	59.00	57.00	58.66
CoTAM	51.00	25.00	36.00	37.33
Logical DA (Ours)	**69.00**	**57.00**	**66.00**	**64.00**

Table 4: Experimental results of data faithfulness.

Data Faithfulness To evaluate the data faithfulness of the data generated by our method, We compare the correctness of the labels generated by different methods against the target labels, which are derived from a more powerful model, GPT-4o, and human annotations. The results, as shown in the table 4, indicate that our method significantly outperforms other methods that use LLM for data generation in terms of label accuracy.

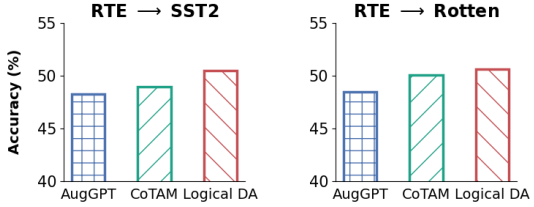

Figure 5: Analysis of task generalization

5.7 Analysis of Model Generalization

To investigate whether our Logical DA improves model generalization, we conduct experiments measuring the cross-task zero-shot performance. The performance of out-of-domain (OOD) data is widely used to verify the model generalization (Xu et al., 2021; Zhong et al., 2022a). Following the approach of (Zhong et al., 2022a), we evaluate model performance on several OOD datasets. Specifically, we fine-tune BERT-BASE model on data generated using different methods (including "AugGPT", "CoTAM", and "Logical DA (ours)") on the RTE task, and then evaluate on other tasks, *i.e.*, SST2 and Rotten tomato. The results are illustrated in Figure 5. We observe that Logical DA consistently outperforms the other counterparts, indicating that our method boosts the performance of models on OOD data.

6 Conclusion

In this study, we propose Logical DA, a multi-agent framework that enhances logical reasoning data augmentation in few-shot settings. The system combines two core functions: generating diverse reasoning data and verifying logical correctness through label validation. By coordinating four specialized agents, it resolves key challenges in existing methods—data repetition, label mismatches, and limited diversity. Experiments demonstrate consistent performance gains in both fine-tuning and in-context learning tasks, advancing language models' reasoning capabilities while reducing reliance on costly manual annotation.

Future work will aim to explore applying our system to other multi-agent frameworks to mitigate the negative impacts caused by contextual logical conflicts in multi-agent collaboration. Also, follow researches can extend this paradigm to multimodal reasoning tasks, and optimize computational efficiency for real-time deployment.

7 Limitations

Despite the significant advancements in generating higher-quality logical reasoning data and enhancing model reasoning capabilities reported in this paper, our work has several potential limitations. Firstly, due to limited computational resources, we have only validated our Logical DA framework on the fine-tuning of pre-trained language models (PLMs) and in-context learning (ICL) of large language models (LLMs). Expanding our experiments to include fine-tuning of LLMs would provide additional evidence of the robustness and generalizability of our approach. Secondly, while Logical DA primarily focuses on logical reasoning tasks, applying it to a wider range of reasoning tasks could significantly increase its utility and impact. Future work will aim to extend this paradigm to broader domains of reasoning. Lastly, the effectiveness of our method depends on the capabilities of underlying LLMs. Limitations in these models could impact the performance of Logical DA.

8 Ethics Statement

Our work involves utilizing large language models to generate high-quality reasoning data, which poses no ethical concerns.

References

Jiaao Chen, Derek Tam, Colin Raffel, Mohit Bansal, and Diyi Yang. 2023. An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11:191–211.

Shuaihang Chen, Yuanxing Liu, Wei Han, Weinan Zhang, and Ting Liu. 2024. A survey on llm-based multi-agent system: Recent advances and new frontiers in application.

Yi Cheng, Wenge Liu, Jian Wang, Chak Tou Leong, Yi Ouyang, Wenjie Li, Xian Wu, and Yefeng Zheng. 2024. Cooper: Coordinating specialized agents towards a complex dialogue goal. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17853–17861.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

Haixing Dai, Zheng Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, W. Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Lichao Sun, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023a. Auggpt: Leveraging chatgpt for text data augmentation.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, et al. 2023b. Chataug: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*, 1(2).

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1679–1705, Bangkok, Thailand. Association for Computational Linguistics.

Zhuoyun Du, Chen Qian, Wei Liu, Zihao Xie, Yifei Wang, Yufan Dang, Weize Chen, and Cheng Yang. 2024. Multi-agent software development through cross-team collaboration. *Preprint*, arXiv:2406.08979.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023a. S3: Social-network simulation system with large language model-empowered agents. *Preprint*, arXiv:2307.14984.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of*

the 2024 Conference on Empirical Methods in Natural Language Processing, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Md. Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. 2024. Mapcoder: Multi-agent code generation for competitive problem solving. *Preprint*, arXiv:2405.11403.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. *arXiv preprint arXiv:2003.02245*.

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

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024a. Emergent world representations: Exploring a sequence model trained on a synthetic task. *Preprint*, arXiv:2210.13382.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024b. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Yichuan Li, Kaize Ding, Jianling Wang, and Kyumin Lee. 2024c. Empowering large language models for textual data augmentation. *arXiv preprint arXiv:2404.17642*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. *Preprint*, arXiv:2305.19118.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2023. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents. *Preprint*, arXiv:2308.05960.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. 2024. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11065–11082, Bangkok, Thailand. Association for Computational Linguistics.

OpenAI. 2024. Learning to reason with llms.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Letian Peng, Yuwei Zhang, and Jingbo Shang. 2024. Controllable data augmentation for few-shot text mining with chain-of-thought attribute manipulation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1–16, Bangkok, Thailand. Association for Computational Linguistics.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15174–15186, Bangkok, Thailand. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 47–57, Dublin, Ireland. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. It's not just size that matters: Small language models are also few-shot learners. *Computing Research Repository*, arXiv:2009.07118.

Murray Shanahan. 2024. Talking about large language models. *Communications of the ACM*, 67(2):68–79.

Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small llms are weak tool learners: A multi-llm agent. *Preprint*, arXiv:2401.07324.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Yi Su, Yunpeng Tai, Yixin Ji, Juntao Li, Bowen Yan, and Min Zhang. 2024. Demonstration augmentation for zero-shot in-context learning. *arXiv preprint arXiv:2406.01224*.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2020. Proofwriter: Generating implications, proofs, and abductive statements over natural language. *arXiv preprint arXiv:2012.13048*.

Xunzhu Tang, Kisub Kim, Yewei Song, Cedric Lothritz, Bei Li, Saad Ezzini, Haoye Tian, Jacques Klein, and Tegawendé F. Bissyandé. 2024. CodeAgent: Autonomous communicative agents for code review.

In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11313, Miami, Florida, USA. Association for Computational Linguistics.

Futing Wang, Jianhao Yan, Yue Zhang, and Tao Lin. 2024a. Elicit: Llm augmentation via external in-context capability. *ArXiv*, abs/2410.09343.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024b. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *Preprint*, arXiv:2402.18272.

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

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models.

Jundong Xu, Hao Fei, Meng Luo, Qian Liu, Liangming Pan, William Yang Wang, Preslav Nakov, Mong-Li Lee, and Wynne Hsu. 2024a. Aristotle: Mastering logical reasoning with a logic-complete decompose-search-resolve framework. *arXiv preprint arXiv:2412.16953*.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024b. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13365, Bangkok, Thailand. Association for Computational Linguistics.

Runxin Xu, Fuli Luo, Zhiyuan Zhang, Chuanqi Tan, Baobao Chang, Songfang Huang, and Fei Huang. 2021. Raise a child in large language model: Towards effective and generalizable fine-tuning. In *EMNLP*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023a. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36:55734–55784.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J. Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2023b. Large language model as attributed training data generator: A tale of diversity and bias. *ArXiv*, abs/2306.15895.

Shengbin Yue, Siyuan Wang, Wei Chen, Xuanjing Huang, and Zhongyu Wei. 2025. Synergistic multi-agent framework with trajectory learning for knowledge-intensive tasks. *Preprint*, arXiv:2407.09893.

Qihuang Zhong, Liang Ding, Li Shen, Peng Mi, Juhua Liu, Bo Du, and Dacheng Tao. 2022a. Improving sharpness-aware minimization with fisher mask for better generalization on language models. In *Findings of EMNLP*.

Wanjun Zhong, Siyuan Wang, Duyu Tang, Zenan Xu, Daya Guo, Yining Chen, Jiahai Wang, Jian Yin, Ming Zhou, and Nan Duan. 2022b. Analytical reasoning of text. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2306–2319.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022a. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Jing Zhou, Yanan Zheng, Jie Tang, Li Jian, and Zhilin Yang. 2022b. FlipDA: Effective and robust data augmentation for few-shot learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8646–8665, Dublin, Ireland. Association for Computational Linguistics.

A Datasets Details

CB: A dataset from FewGLUE (Schick and Schütze, 2020), containing examples with two sentences: a premise p and a hypothesis h. The task is to determine whether the relationship between the premise and the hypothesis is one of entailment, contradiction, or neutrality.

RTE: A dataset from FewGLUE (Schick and Schütze, 2020), also containing examples with two sentences: a premise p and a hypothesis h. CB is specifically designed to test common sense and reasoning abilities, often requiring deeper understanding and inference. RTE, on the other hand, focuses more on whether the premise can directly entail the hypothesis, typically involving more straightforward logical relationships.

ProofWriter (Tafjord et al., 2020): A widely used dataset for deductive logical reasoning. The problems are presented in a more natural language format. We utilize the open-world assumption (OWA) subset, where each example consists of a (problem, goal) pair, and the label is one of $\{Proved, Disproved, Unknown\}$.

LogicalDeduction: A challenging logical reasoning task from the BigBench (Srivastava et al.,

2022) collaborative benchmark. The problems primarily involve deducing the order of a sequence of objects based on a minimal set of conditions.

AR-LSAT (Zhong et al., 2022b): This dataset compiles all analytical logic reasoning questions from the Law School Admission Test (LSAT) administered between 1991 and 2016.

FOLIO (Han et al., 2024): A challenging expert-written dataset designed for logical reasoning tasks. The questions are closely aligned with real-world knowledge, use highly natural language, and require complex first-order logic reasoning to solve.

B Baseline Details

FlipDA++ (Zhou et al., 2022b): FlipDA is a traditional method for data augmentation using a well-tuned T5 model to switch labels. The sentence and the new label are given to T5, which then masks and fills in parts of the sentence based on the new label to change its meaning. Since the original FlipDA needs a lot of labeled data, which is not suitable for few-shot learning, FlipDA++ was created (Peng et al., 2024), which works by telling the LLM to replace parts of the sentence with the new label.

AugGPT (Dai et al., 2023a): an approach that rephrases each sentence in the training samples into multiple conceptually similar but semantically different samples.

AttrPrompt (Yu et al., 2023b): a framework designed to generate diverse and less biased training data for NLP tasks by using attributed prompts. It leverages Large Language Models (LLMs) to generate training data with specific attributes, aiming to improve data diversity and reduce systemic bias.

CoT Attribute Manipulation (CoTAM) (Peng et al., 2024): an approach that generates new data from existing examples by only tweaking the user-provided, task-specific attribute, e.g., sentiment polarity or topic in movie reviews.

CoT Data Augmentation (CoTDA) (Peng et al., 2024): an augmentation variant of CoTAM that applies a similar CoT for conventional augmentation. Instead of directly asking for augmentation, CoTDA let the LLM follow our proposed CoT and propose a methodology to write a sentence with the same attributes as the input sentence.

Self-LLMDA (Li et al., 2024c): a framework that automates augmentation instruction generation and selection, facilitating LLM to generate task-specific augmented data.