Dually Self-Improved Counterfactual Data Augmentation Using Large Language Model

Luhao Zhang[1,*], **Xinyu Zhang**[1,*], **Linmei Hu**[1,†], **Dandan Song**[1], **Liqiang Nie**[2]
[1] Beijing Institute of Technology, Beijing, China,
[2] Harbin Institute of Technology, Shenzhen, China
{zhangluhao, hulinmei, songdandan}@bit.edu.cn,
{xyzhang0105, nieliqiang}@gmail.com

Abstract

Counterfactual data augmentation, which generates minimally edited tokens to alter labels, has become a key approach to improving model robustness in natural language processing. It is usually implemented by first identifying the causal terms and then modifying these terms to create counterfactual candidates. The emergence of large language models (LLMs) has effectively facilitated the task of counterfactual data augmentation. However, existing LLM-based approaches still face some challenges in 1) accurately extracting the task-specific causal terms, and 2) the quality of LLM-generated counterfacts. To address the issues, we propose a dually self-improved counterfactual data augmentation method using LLM. On the one hand, we design a self-improved strategy employing the attention distribution of the task model to identify the task-specific causal terms, which is lightweight and task-specific. On the other hand, a second self-improved strategy based on direct preference optimization is utilized to refine LLM-generated counterfacts, achieving high-quality counterfacts. Finally, a balanced loss preventing over-emphasis on augmentated data is proposed to retrain the task model on the fusion of existing data and generated counterfacts. Extensive experiments on multiple benchmarks demonstrate the effectiveness of our proposed method in generating high-quality counterfacts for improving task performance.

1 Introduction

In the complex realm of machine learning and NLP, imbalance, and biases prevalent in real-world training data continue to be an arduous challenge for robust model development. Traditional data augmentation suffers from the issue of spurious association when alleviating these issues (Chen et al., 2021). In recent years, generating counterfactual augmented

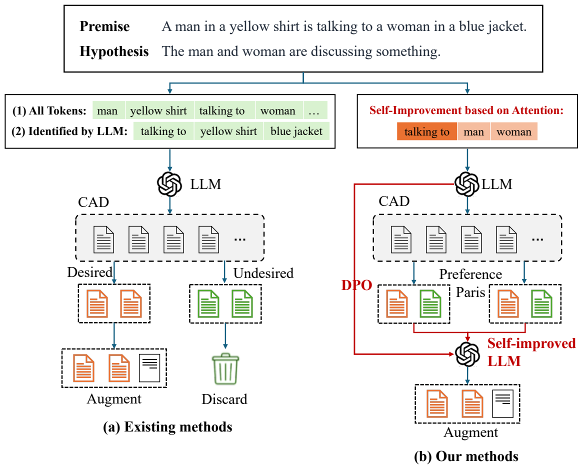

Figure 1: Introduction of Counterfactual Data Augmentation.

data (CAD) (Kaushik et al., 2020), introducing minimal modifications to the data through additions, replacements, or deletions to flip the label, has been widely attempted in many tasks (Liu et al., 2021a). Target task models trained with large-scale counterfacts can learn better representations and effects of casual terms, which facilitates task performance improvements and enables robust generalization.

Typically, counterfactual data augmentation involves three steps: (1) identifying important tokens (known as causal terms) that can flip the labels, (2) minimally editing these terms to create counterfactual candidates, and (3) retraining the model on the fusion data of existing data and augmented data. For example, as shown in Figure 1, in NLI task, through modifying the identified casual term "*talking to*" to "*walking with*" for the given example, we flip the original label from "*Entailment*" to "*contradiction*", obtaining a counterfact.

However, it is non-trivial to obtain high-quality counterfacts. Early works (Gardner et al., 2020; Kaushik et al., 2020) relied on human experts to annotate counterfactual examples, which is not easily scalable. Therefore, researchers have been explor-

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 5216–5227
July 27 - August 1, 2025

ing automatic methods for counterfactual generation using neural networks (Chen et al., 2021). Recently, AutoCAD (Wen et al., 2022) has attempted to leverage generative language models, such as T5 (Raffel et al., 2020), for controllable text generation. However, due to the limited comprehension and generation capabilities of these language models, the quality of the generated data remains constrained. The advent of LLMs has driven significant progress across various NLP tasks (Chen, 2024), researchers have focused on designing effective prompts to leverage the advanced comprehension and generation abilities of LLMs for directly generating desired counterfacts (Chen et al., 2023; Dixit et al., 2022; Nguyen et al., 2024).

Despite the promising advancements, research on LLM-based counterfactual data augmentation still faces several challenges. (1) How to extract causal terms specific to the task accurately? Existing works either exploited all spans obtained through sentence splitting (Chen et al., 2023), or directly prompted LLMs (Li et al., 2024) to identify causal terms. All of these methods suffer from the inaccurate casual terms specific to the task. (2) How to enhance the quality of LLM-generated CAD by modifying the causal terms? Those LLM-based approaches typically employ LLMs to rewrite causal terms and then select the desired counterfacts with a score function. However, the quality of the generated counterfacts is still suboptimal since the LLM is not specially optimized for generating CAD, and the low-scored data is also not fully leveraged.

In this paper, to address the above issues, we propose a **d**ually self-**i**mproved **c**oun**t**erfactual data augmentation method using LLMs (**DICT**). On one hand,

as the attention mechanism offers insights into the causal relationships between texts and their labels (Nauta et al., 2019), we design a self-improved strategy based on the attention distribution of the target task model to identify causal terms, a lightweight and task-specific approach. As shown in Figure 1, the terms with larger attention of the target task model are more critical for the NLI label, while existing methods suffer from the accuracy of the identified causal terms and may introduce noise.

On the other hand, to further improve the quality of CAD, we propose an additional self-improved strategy based on direct preference optimization (DPO) to refine itself. Specifically, after generating preliminary counterfacts, we construct the preference pairs based on the score function for DPO.

Finally, through simple filtering and fusion, we retrain the task model on the fused data, using a balanced loss function to avoid over-emphasis on augmented data. Overall, our contributions can be summarized as follows:

- We propose a dually self-improved counterfactual data augmentation, improving the counterfactual data augmentation framework depending on the task model and LLM themselves, without external tools to identify casual terms or human annotation for fine-tuning LLMs.

- Our proposed DICT improves the extraction of task-specific causal terms through attention mechanisms and further enhances the CAD generation of LLMs using DPO. Additionally, a novel balanced loss is introduced to retrain the task model on the fused data, effectively preventing excessive augmentation.

- Extensive experiments across multiple benchmarks demonstrate that DICT significantly outperforms the state-of-art annual and automatic CAD generation methods across all metrics.

2 Related Work

Counterfactual Data Augmentation. Generating fluent textual CAD are required to follow some principles, including: (1) minimal edits, (2) fluency, creativity, and diversity, and (3) adhering to task-specific rules (Wang et al., 2024). However, these requirements have been proved challenging . Early, Kaushik et al. and Gardner et al. (2020) employ human annotators to create counterfacts by manually rewriting the original data. Obviously, manual rewrites are not only time-consuming and expensive but also may exacerbate existing spurious features. To alleviate the mentioned issues, Tokpo and Calders (2024) rely on additional word dictionaries to select casual terms, which is inaccurate and difficult to be generalized. Further, researchers (Madaan et al., 2021; Ross et al., 2021; Wen et al., 2022) proposed using advanced text generation models, such as T5 (Raffel et al., 2020), to generate CAD. Due to the limited comprehension and generation capabilities of previous generative language models, the quality of the generated data remains constrained. Additionally, some works (Liu et al., 2021b; Zeng et al., 2020) consider the task-specific issue when generating CAD, which cannot generalize to other tasks. For example, TCWR (Liu et al., 2021b) considers the symmetry between source and target sequences in Natural Machine Translation when generating CAD.

LLM-based Counterfactual Data Augmentation. LLMs have shown remarkable proficiency in synthesizing natural languages for downstream tasks. Leveraging the powerful generative ability of LLMs to automatically generate counterfacts has recently attracted considerable attention (Liu et al., 2020a). DISCO (Chen et al., 2023) prompts GPT3 (Brown et al., 2020) to generate phrasal perturbations for automatically generating CAD at scale. Nguyen et al. (2024) and Li et al. (2024) investigated the strengths and weaknesses of LLMs as generators comprehensively, instructing LLMs to identify casual terms and generate counterfacts.

However, despite the significant advancements, the quality of counterfactual augmented data with LLMs still remains to be improved since LLMs are not specially trained for CAD generation. Our work bridges this gap by designing a dually self-improved method to enhance both the extraction of the specific causal terms and the generation of CAD (modifying the causal terms) with LLMs.

3 Preliminaries

We implement counterfactual data augmentation on the Natural Language Inference (NLI) task, referring to determining the relationship between a given premise sentence and a hypothesis sentence (Hosseini et al., 2024). Formally, given an input premise-hypothesis pair $\langle \mathrm{P}_i, \mathrm{H}_i \rangle$ and its ground-truth label l_i, where $\mathrm{P}_i = \{t_1, t_2, \cdots, t_m\}$, $t_j = \{w_1, \cdots, w_n\}$ represents a token that consists of n words [1], and m is the number of tokens. $l_i \in \{\text{Entailment}, \text{Contradiction}, \text{Neutral}\}$, the task aims to produce a counterfactual example $\langle \hat{\mathrm{P}}_i, \mathrm{H}_i \rangle$ that flips the origin label l to a desired label $\hat{l}_i$, $\hat{l}_i \neq l_i$, through perturbing parts of the premise P_i. When the original premise P_i is altered into counterfactual $\hat{\mathrm{P}}_i$, minimal changes are required. Here, casual terms are denoted as $\mathrm{C}_i = \{c_1, \cdots, c_k\}$, where each c_j corresponds to a token t_j extracted from P_i. After CAD generation, the performance is evaluated through a baseline NLI model $\mathbb{M}$, such as RoBERTa (Liu et al., 2020b).

4 Our Proposed Model

In this section, we detail our proposed dually self-improved counterfactual data augmentation method using a large language model (DICT).

As shown in Figure 2, our model consists of three stages: 1) self-improved casual terms identification, 2) self-improved CAD generation, 3) retraining. First, we design a self-improvement strategy leveraging the attention distribution of the task model to enhance the identification of causal terms. Second, we further propose to utilize a self-improved LLM based on DPO to refine the CAD generation by modifying the causal terms. Finally, after filtering and fusing the generated counterfacts, we retrain the task model with a balanced loss function, avoiding over-augmentation. In this way, we improve the task model performance with our generated augmented counterfactual data.

4.1 Self-improved Casual Terms Identification

Casual terms capture the effective features implied in sentences. Therefore, identifying causal terms is the crucial first step of counterfactual data augmentation. To achieve this, we propose a self-improved causal term identification method based on the attention distribution of the task model. Different attention layers can be seen as a hierarchy that gradually refines the context of the input sequence; the higher layers focus on more abstract semantic understanding (Clark et al., 2019; Gillioz et al., 2020). Therefore, given the task model $\mathbb{M}$ trained on the original dataset and a premise-hypothesis sample $\langle \mathrm{P}_i, \mathrm{H}_i \rangle$, we utilize the last attention layer of the task model to compute the attention score α_{w_i} on each word w_i of premise P_i under its label l_i:

$$\alpha_{w_i} = \text{Attention}_{\mathbb{M}}(l_i | \mathrm{P}_i, \mathrm{H}_i), \tag{1}$$

where $\text{Attention}_{\mathbb{M}}$ is the last attention layer embedded in the task model $\mathbb{M}$. Then, the attention score α_{t_j} on each token t_j is calculated as

$$\alpha_{t_j} = \text{Average}(\alpha_{w_1}, \cdots, \alpha_{w_n}), \tag{2}$$

where Average is a mean-pooling layer. In this way, we obtain the attention weights of tokens. Finally, tokens are sorted in descending order based on the attention score α_{t_j}, and top K ($K = 3$ in this paper) tokens are selected as the final causal terms C_i.

[1] We split sentences into tokens through Flair (Akbik et al., 2018).

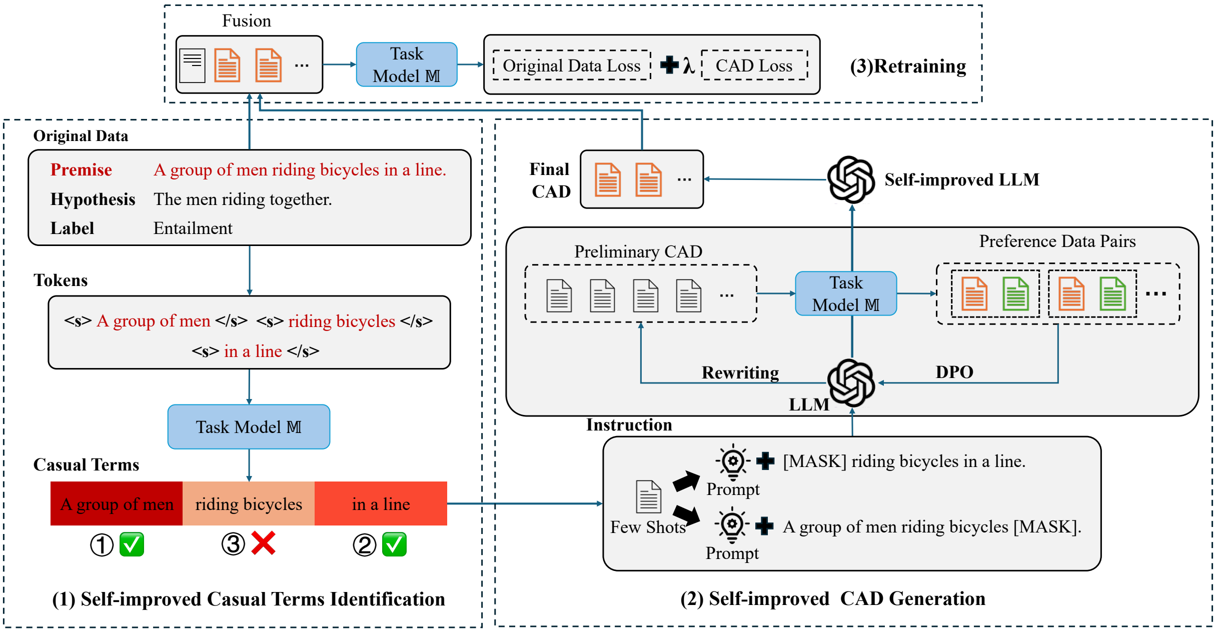

Figure 2: The architecture of our proposed DICT.

4.2 Self-improved CAD Generation

With the identified causal terms and original sentence pairs, we propose a self-improved LLM based on DPO to modify causal terms, thereby flipping the label and generating CAD.

First, each casual term C_i is replaced with a mask token [MASK] individually to obtain K sentences to be rewritten. Then, for each sentence, we instruct an LLM to alter the [MASK] into certain tokens for flipping the original label l_i of the $\langle \mathrm{P}_i, \mathrm{H}_i \rangle$ into a specific label $\hat{l}_i$. To achieve this, the prompt (shown in Appendix A.1) is designed to instruct an LLM to generate CAD. Note that, for each causal term, we employ an over-generation strategy to generate multiple corresponding candidate counterfacts $\{\hat{\mathrm{P}}_i^1, \cdots, \hat{\mathrm{P}}_i^o\}$ by rephrasing the causal terms. Afterward, all the candidate counterfacts are scored via the predicted probability shift of the target label $\hat{l}_i$ based on the task model $\mathbb{M}$:

$$\delta_j = p(\hat{l}_i | \hat{\mathrm{P}}_i^j, \mathrm{H}_i) - p(\hat{l}_i | \mathrm{P}_i, \mathrm{H}_i). \tag{3}$$

Instead of directly using the filtered results by the calculated scores δ, we design another self-improved strategy based on DPO to achieve self-improved LLM for generating higher-quality candidate counterfacts. Specifically, for each causal term in C, we choose the corresponding generated candidate counterfact (by modifying the causal term) with the highest score δ as the accepted example $\hat{\mathrm{P}}_i^1$, and a random one with $\delta < \gamma$ as a rejected example $\hat{\mathrm{P}}_i^2$, where γ is the threshold and set to 0.7 in this work. Formally, by forming the two samples, the entire preference pair data are denoted as:

$$\mathbb{P} = \{(\mathrm{P}_i, \hat{\mathrm{P}}_i^1, \hat{\mathrm{P}}_i^2))\}_{i=1}^N. \tag{4}$$

Self-Improved LLM based on DPO. As defined previously, we prefer the counterfact $\hat{\mathrm{P}}_i^1$ to $\hat{\mathrm{P}}_i^2$ given an input P_i. To enable the LLM to learn this desired preference, DPO is employed to refine the LLM using the preference pairs. Formally, the preference probability is first predicted as follows:

$$r(\mathrm{P}_i, \hat{\mathrm{P}}_i) = \beta \log \frac{\pi_r(\hat{\mathrm{P}}_i | \mathrm{P}_i)}{\pi_{ref}(\hat{\mathrm{P}}_i | \mathrm{P}_i)} + \beta \log \mathbb{Z}(\mathrm{P}_i), \tag{5}$$

$$p(\hat{\mathrm{P}}_i^1 > \hat{\mathrm{P}}_i^2 | \mathrm{P}_i) = \frac{1}{1 + e^{r(\mathrm{P}_i, \hat{\mathrm{P}}_i^1) - r(\mathrm{P}_i, \hat{\mathrm{P}}_i^2)}}, \tag{6}$$

where $r(\mathrm{P}_i, \hat{\mathrm{P}}_i)$ is the reward function with the input of any generated counterfact $\hat{\mathrm{P}}_i$ and its origin text P_i, π_r and π_{ref} are respectively the corresponding optimal policy and the reference policy, $\mathbb{Z}(\cdot)$ is the partition function and β is a parameter controlling the deviation from the reference policy.

Then, LLMs can be directly optimized with preference probabilities (DPO) using the following binary cross-entropy loss function:

$$\begin{aligned} L(\pi) = -\sum_{\mathbb{P}} [&p(\hat{\mathrm{P}}_i^1 > \hat{\mathrm{P}}_i^2 | \mathrm{P}_i) \log \pi_r(\hat{\mathrm{P}}_i^1 | \mathrm{P}_i)) \\ &+ (1 - p(\hat{\mathrm{P}}_i^2 > \hat{\mathrm{P}}_i^1 | \mathrm{P}_i)) \log (1 - \pi_r(\hat{\mathrm{P}}_i^1 | \mathrm{P}_i))]. \end{aligned} \tag{7}$$

Subsequently, we apply the self-improved LLM to generate higher-quality CAD. The generated candidate counterfacts are further filtered based on the aforementioned probability shift score δ to ensure the data quality (i.e., δ is above the threshold γ).

4.3 Retraining

Finally, we fuse the filtered CAD with the original data and retrain the task model to improve the task performance. As the scale of counterfactual data grows, we observe that the task model may overly focus on the counterfactual data while overlooking the original data. Therefore, during the retraining, a penalty factor λ is used to balance the original data and the augmented data, improving the robustness of the model while preventing over-emphasis on the augmentation. The loss function is calculated through the cross entropy:

$$L = \mathbb{CE}(p(l|\mathrm{P},\mathrm{H}),l) + \lambda \cdot \mathbb{CE}(p(\hat{l}|\hat{\mathrm{P}},\mathrm{H}),\hat{l}), \quad (8)$$

where $\mathbb{CE}$ is the cross entropy function, and λ is the balance factor.

5 Experiments

5.1 Datasets

We evaluate the overall performance on NLI tasks over three benchmarks, including two in-domain subsets from SNLI(Bowman et al., 2015) and MNLI (Williams et al., 2018). In the following, we detail each dataset.

- SNLI (Bowman et al., 2015). The Stanford Natural Language Inference (SNLI) corpus, derived from only one domain, is a collection of sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral. The first subset SNLI-1, following (Wen et al., 2022), consists of an ambiguous part of SNLI. It contains 20,000 examples for training, 4,800 for validation, and 4,800 for testing. To further evaluate the performance, we extracted a larger-scale examples randomly from the original SNLI corpus, consisting of 87,208, 18,688, and 18,688 pairs for training, validation, and testing respectively.

- MNLI (Williams et al., 2018). Multi-genre NLI corpus (MNLI), including two different test sets MNLI-matched (MNLI-m) and MNLI-mismatched (MNLI-mm) [2], is a multiple out-of-domain and challenge benchmark to measure the generalization of the model after data augmentation. It contains 392,702 pairs in the train set, 9,815 in the MNLI-m test set, and 9,796 pairs in the MNLI-mm test set.

5.2 Baselines

We compare our model with the state-of-the-art baselines:

- RoBERTa-large (Liu et al., 2019). A robustly optimized SOTA transformer model pre-trained on a large corpus. It is used as the target task model to be augmented.

- HumanCAD (Kaushik et al., 2020). A manual set of CAD for NLI, obtained by human annotators rewriting a subset of SNLI. We append them into original benchmarks and evaluate the performance following (Wen et al., 2022).

- AutoCAD (Wen et al., 2022). A fully automatic CAD generation framework with the generative language model T5.

- DISCO (Chen et al., 2023). A counterfactual knowledge distillation approach with LLMs. It leverages all spans as causal terms for CAD generation and filters out unqualified generated data using a SOTA task-specific model.

- LLMCF (Li et al., 2024). A CoT-based method that prompts LLMs to identify causal terms and produce CAD. To ensure a fair comparison, we adopt the task model to filter the generated CAD, as we do in our DICT.

Note that, for fair comparison, all baseline methods and our DICT use the same task model RoBERTa-large and aim to improve the task model with the generated counterfactual augmented data.

5.3 Experimental Settings

For SNLI-1, we perform counterfactual augmentation on each sample. Due to the large scale of the SNLI-2 and the MNLI, we sampled subsets of a fixed size for counterfactual augmentation, including 50,000 examples from the training set. Following (Chen et al., 2023), we measure the consistency

[2]The details can be found in the website https://cims.nyu.edu/ sbowman/multinli/

Dataset	SNLI-1			SNLI-2			MNLI-m			MNLI-mm		
Metric(%)	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
RoBERTa-large	61.36	59.77	58.29	87.92	86.76	86.82	87.38	87.23	87.27	87.06	86.92	86.97
Human-CAD	60.90	62.27	61.26	87.57	87.51	87.51	87.17	86.92	86.85	87.30	87.06	87.10
AutoCAD	57.08	58.58	57.48	87.37	87.35	87.36	87.52	87.33	87.41	87.44	87.32	87.37
DISCO-7B	59.50	61.18	59.26	87.80	87.73	87.75	87.76	87.77	87.76	87.56	87.50	87.54
LLMCF-7B	61.17	61.43	60.24	88.43	87.39	87.65	87.80	87.66	87.71	87.70	87.57	87.62
LLMCF-14B	63.15	63.43	62.84	88.82	88.79	88.79	88.89	88.73	88.84	88.72	88.66	88.68
DICT-7B	62.38	62.39	61.37	88.63	87.78	87.89	88.23	88.07	88.15	88.12	87.85	87.91
DICT-14B	**65.10**	**65.08**	**64.89**	**89.42**	**89.51**	**89.47**	**89.44**	**89.33**	**89.36**	**89.28**	**89.25**	**89.26**

Table 1: Performance comparison of different methods over Precision, Recall and F1 score, where 7B and 14B means Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct as the base LLM respectively.

of model performances on the original and counterfactual test examples. We sample 2,000 examples from the test sets respectively for generating CAD . In terms of LLM-based models, we use Qwen2.5-7B-Instruct and Qwen2.5-14B-Instruct (Yang et al., 2024; Team, 2024) as the base LLMs. The prompt for instructing LLMs follows (Chen et al., 2023), ensuring a fair comparison and minimizing the impact of prompt variations on the generated counterfacts. The RoBERTa-large model is trained on all basic and augmented datasets with a learning rate of 1e-5 for 3 epochs. The size of obtained preference pairs is approximately 25,000 across all the datasets. For the DPO process, we set the number of epochs to 1. The penalty factors λ in the loss function are 0.4 and 0.6 for DICT-7B and DICT-14B, respectively. All the reported results of our DICT are the average results of three runs.

5.4 Overall Performance

To assess the overall performance, we perform counterfactual data augmentation on the training data and conduct evaluation on the original test set. As shown in Table 1, we report Precision (P), Recall (R) and F1-score (F1) respectively on all datasets to evaluate the overall performance of CAD methods. Concretely, the task model RoBERTa is trained on the fusion of the generated counterfacts and the original data, and evaluated on the original test data. It can be observed that: (1) all counterfactual data augmentation methods prove effective in most cases. However, due to the higher ambiguity and difficulty of SNLI-1, AutoCAD slightly weakens the model performance. (2) LLM-based methods outperform AutoCAD in most cases, indicating the powerful comprehension and generation capabilities of LLMs. (3) Our proposed model DICT achieves the best results across both 7B and 14B settings, especially on the more challenging SNLI-1 dataset and the out-of-domain MNLI-mm dataset. It demonstrates the robustness and effectiveness of our proposed DICT. (4) Both LLMCF and DICT exhibit significant performance improvements as the LLM scale increases, demonstrating that larger models can capture more complex causal relationships and generate higher-quality counterfactual data, leading to better task performance. Note that, DICT performs best in all cases. We believe the reason is that DICT with dual self-improvement can accurately identify the task-specific causal terms and generate higher-quality counterfacts. To further assess the generalizability of our method, we also extend DICT to the sentiment analysis task and demonstrate the effectiveness of DICT, with the results presented in Appendix B.

Method	FR	ACC_δ
Auto-CAD	0.46	0.59
DISCO-7B	0.61	0.77
LLMCF-7B	0.60	0.81
LLMCF-14B	0.71	0.83
DICT-7B	0.80	0.84
DICT-14B	**0.82**	**0.87**

Table 2: Evaluation of the quality of generated counterfacts.

5.5 The Quality of Generated Counterfacts

Following (Nguyen et al., 2024; Chen et al., 2023), we use the filp rate (FR) and the counterfactual accuracy ACC_δ to evaluate the quality of generated counterfacts on SNLI-1. Specifically, FR quantifies how effectively a method can alter the labels of instances and a higher FR indicates more confident

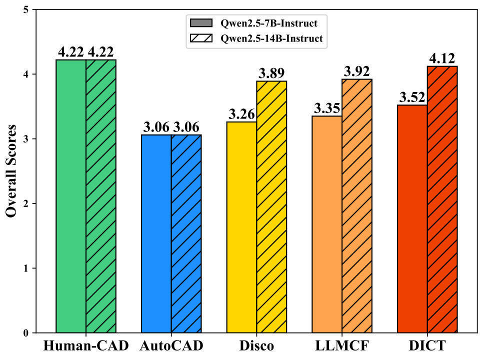

Figure 3: Evaluated Results with GPT4 Over Qwen2.5-7B and Qwen2.5-14B Respectively on SNLI-1.

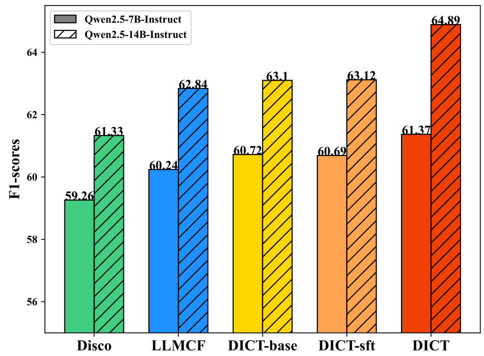

Figure 4: Ablation study over Qwen2.5-7B and Qwen2.5-14B on SNLI-1.

and impactful context modifications. FR is defined as:

$$\mathrm{FR} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[p(\hat{l}_i|\hat{\mathrm{P}}_i, \mathrm{H}_i) = \hat{l}_i], \tag{9}$$

, where $\mathbb{I}$ is an indicator function that outputs 1 if the predicted label of a counterfact matches its desired label. The FR is evaluated using the counterfactual augmentation results on the training set, where the probability $p(\hat{l}_i|\hat{\mathrm{P}}_i, \mathrm{H}_i)$ is computed using the task model.

The counterfactual accuracy ACC_δ is used to measure the consistency of the DICT's performance on original and counterfactual examples of test data, and is defined as:

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{I}[p(\hat{l}_i|\hat{\mathrm{P}}_i, \mathrm{H}_i) = \hat{l}_i \wedge p(l_i|\mathrm{P}_i, \mathrm{H}_i) = l_i], \tag{10}$$

where $\mathbb{I}$ indicts 1 only when the model correctly predicts the original and counterfactual examples. All probabilities are computed using the augmented task model on the test set and their corresponding counterfacts. Therefore, test samples linked to corresponding counterfactual examples are preserved.

As shown in Table 2, our model achieves the best performance on both FR and ACC_δ. DICT-14B increases the FR by around 15% compared to LLMCF-14B, demonstrating that DICT effectively produces a larger quantity of high-confidence counterfacts. Additionally, the results on ACC_δ also highlight that our DICT exhibits better consistency and generalization.

Evaluation with GPT-4. GPT-4 is a reliable evaluator for accessing the quality of CAD, as demonstrated in (Nguyen et al., 2024; Liu et al., 2023; Azaria et al., 2024). Accordingly, we select 1,000 samples randomly from SNLI-1 for all methods and use GPT-4 to assign an overall score (on a 5-point scale) to them from three aspects, including fluency, realism, and conciseness. The utilized instruction is detailed in Appendix A.2. As shown in 3, compared to Auto-CAD that employs traditional generative language models, LLM-based models achieve higher scores obviously. Despite that all model-based methods fall short of Human-CAD, our DICT still achieves superior performance over Human-CAD. Simultaneously, as the scale of the large models increases, the scores show significant improvements.

5.6 Ablation Study

In order to verify the effectiveness of different modules of our model, we design two variant models:

- **DICT-base** removes the self-improved generator and use a basic LLM to produce CAD.
- **DICT-sft** replaces the DPO strategy with supervised fine-tuning (SFT). Instead of improving the LLM on preference data pairs, it just employs the preferred parts.

They are both compared to LLM-based methods on SNLI-1 dataset with Qwen2.5-7B and Qwen2.5-14B respectively. As shown in Figure 4, we report F1-scores as evaluated results. Without a self-improvement generator, the performances are still better than both DISCO and LLMCF. It demonstrates that our self-improved identifier can identifying specific casual terms that are crucial for generating CAD. If we replace DPO with SFT as our self-improved strategy of the generator, the performances of DICT-sft decrease by 0.5% and 1.77% over Qwen2.5-7B and Qwen2.5-14B respectively. It indicates the necessity of designing a self-

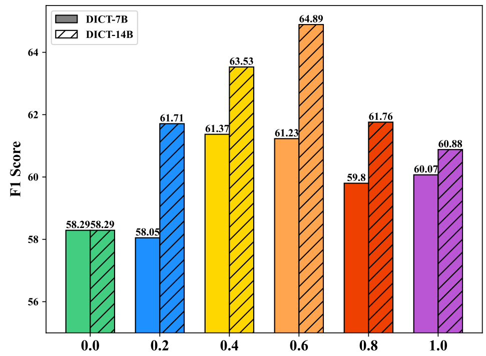

Figure 5: The impact of Hyperparameter λ for DICT-7B and DICT-14B on SNLI-1.

improved strategy to enhance the LLM's rewriting capability of CAD. We also find that the performances of DICT-sft increase in-obviously compared to DICT-base. The reason may be that without the constraint of negative samples, the optimization space of the LLM becomes more complicated in our task. It is assumed that there should be more high-confidence CAD to train the LLM better with SFT. Additionally, as the parameter scale of the LLM increases, the performance of all methods improves significantly, further validating that larger models can generate higher-quality counterfactual data.

5.7 HyperParameter Experiments

We validate the impact of different hyperparameters λ within $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ on preventing over-emphasis on augmented data. When λ is equal to 0, the DICT degenerates to the basic model RoBERTa. As shown in Figure 5, when λ is relatively small (e.g., 0.2 or below), the model primarily focuses on original data, limiting the benefits of counterfactual data augmentation. Conversely, when λ is too high (e.g., 1.0), the model heavily emphasizes CAD, degrading the performance. Optimal results are observed within the range of $\lambda \in [0.4, 0.6]$ for both DICT-7B and DICT-14B, where the balance between original and generated counterfactual data contributes to improving the performance.

5.8 Case Study

Figure 6 shows counterfual examples from SNLI-1. In Case 1, key tokens in premises like "*earth globe*" and "*purple*" significantly influence the relationship with the hypothesis, namely the NLI label. Our DICT can successfully extract these tokens as causal terms for modifying to flip the NLI label. This step ensures that the counterfactual generation is grounded in the critical linguistic features. Thus, the generated coungterfacts are of high quality.

6 Conclusion

In this paper, we address the challenges in LLM-based counterfactual data augmentation by introducing the proposed DICT method, a dually self-improved counterfactual data augmentation approach using LLM. Specifically, we first introduce a lightweight and task-specific causal term identification strategy that leverages the attention distribution of the task model for self-improvement. This approach effectively captures causal terms by interpreting the attention scores, overcoming the limitations of LLMs in accurately identifying specific causal terms. Second, we propose a self-improved counterfactual generator that modifies the causal terms to flip the label based on DPO. By constructing preference data pairs from the preliminary generated counterfacts, we refine the LLM with DPO, ensuring higher-quality counterfactual generation. Our experimental results demonstrate that DICT outperforms existing LLM-based counterfactual data augmentation methods across various NLI datasets, achieving superior performance in terms of both accuracy and robustness. Additionally, we observe that increasing the LLM's parameter scale further boosts the performance, highlighting the scalability and effectiveness of our proposed method.

Furthermore, our DICT can be directly applied to various NLP tasks such as relation extraction, which we will explore in future work.

7 Limitation

While DICT demonstrates strong performance, it is inherently dependent on the capabilities of the underlying large language models (LLMs). This dependence means that DICT's effectiveness can vary across different LLM architectures and versions, highlighting the need for a strong LLM backbone to ensure reliable outcomes.

Acknowledgements

This work was supported by the National Science Foundation of China (No. 62276029), Technical Field Foundation (No. 2023-JCJQ-JJ-0747), Beijing Institute of Technology Research Fund Program for Young Scholars (No.6120220261),

	Case 1	Case 2	Case 3
Original Premise	Two people are holding a ***large upside-down earth globe***, about 4‘ in diameter, and a child appears to be jumping over Antarctica.	A woman wearing orange ***looking upward.***	An oriental girl is ***searching*** a baby in her arms.
Original Hypothesis	The earth globe is purple.	A woman gazes at her shoes.	The girl is looking for her baby brother.
Original Label	Contradiction	Contradiction	Entailment
Counterfactual Premise	Two people are holding ***a large purple earth globe***, about 4‘ in diameter, and a child appears to be jumping over Antarctica.	A woman wearing orange ***looking down at her orange high heels***.	An oriental girl is ***holding*** a baby in her arms.
Flipped Label	Entailment	Entailment	Contradiction

Figure 6: Counterfactual examples from SNLI-1 generated by our DICT.

and CIPSC-SMP-Zhipu Large Model Cross-Disciplinary Fund.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Amos Azaria, Rina Azoulay, and Shulamit Reches. 2024. Chatgpt is a remarkable tool—for experts. *Data Intelligence*, 6(1):240–296.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901.

Hao Chen, Rui Xia, and Jianfei Yu. 2021. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Huajun Chen. 2024. Large knowledge model: Perspectives and challenges. *Data Intelligence*, 6(3):587–620.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. DISCO: Distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Anthony Gillioz, Jacky Casas, Elena Mugellini, and Omar Abou Khaled. 2020. Overview of the transformer-based models for nlp tasks. In *2020 15th Conference on computer science and information systems*, pages 179–183. IEEE.

Mohammad Javad Hosseini, Andrey Petrov, Alex Fabrikant, and Annie Louis. 2024. A synthetic data approach for domain generalization of nli models. *arXiv preprint arXiv:2402.12368*.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Divyansh Kaushik, Amrith Setlur, Eduard H Hovy, and Zachary Chase Lipton. Explaining the efficacy of counterfactually augmented data. In *International Conference on Learning Representations*.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2024. Prompting large language models for counterfactual generation: An empirical study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language*

Resources and Evaluation (LREC-COLING 2024), pages 13201–13221.

Qi Liu, Matt Kusner, and Phil Blunsom. 2021a. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Qi Liu, Matt Kusner, and Phil Blunsom. 2021b. Counterfactual data augmentation for neural machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 187–197.

Tianyu Liu, Zheng Xin, Baobao Chang, and Zhifang Sui. 2020a. HypoNLI: Exploring the artificial patterns of hypothesis-only bias in natural language inference. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6852–6860.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Roberta: A robustly optimized bert pretraining approach.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13516–13524.

Meike Nauta, Doina Bucur, and Christin Seifert. 2019. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, pages 312–340.

Van Bach Nguyen, Paul Youssef, Christin Seifert, and Jörg Schlötterer. 2024. LLMs for generating and evaluating counterfactuals: A comprehensive study. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14809–14824.

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

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Ewoenam Kwaku Tokpo and Toon Calders. 2024. Fairflow: An automated approach to model-based counterfactual data augmentation for nlp. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 160–176.

Yongjie Wang, Xiaoqi Qiu, Yu Yue, Xu Guo, Zhiwei Zeng, Yuhong Feng, and Zhiqi Shen. 2024. A survey on natural language counterfactual generation. *arXiv preprint arXiv:2407.03993*.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. AutoCAD: Automatically generate counterfactuals for mitigating shortcut learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2302–2317.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7270–7280.

A Instruction

A.1 Instruction for CAD Generation

Taking the NLI task as the example, we design the following instruction for generating counterfacts:

> Given the conclusion, the statement, and what you know about the world, fill in the [MASK] to complete the statement so that the conclusion is absolutely true based on the statement. Do not repeat the original statement or the conclusion when completing the statement. Be creative and specific, yet brief and concise.
> **Statement**: A juggling street performer [MASK]. **Conclusion**: A street performer does acrobatic tricks for onlookers. [MASK] should be:
> is doing flips for people who are watching
> **Statement**: A man jumps highly in front [MASK]. **Conclusion**: A man dove into the water. [MASK] should be:
> of a large diving pool
> **Statement**: Two children wearing helmets [MASK]. **Conclusion**: The children are on an exercise bike. [MASK] should be:
> are pedaling as if they are riding a bicycle, but without having to go anywhere.
> **Statement**: A cashier at [MASK]. **Conclusion**: A cashier is currently working. [MASK] should be:

A.2 Instruction for GPT-4

The detailed instruction for using GPT4 as an evaluator is:

> Assuming you are a manual annotator, please evaluate the following counterfactual data based on the following criteria, each on a scale from 1 to 5, where 5 is the best:
> Fluency: How natural and grammatically correct is the generated text?
> Realism: How plausible and contextually appropriate is the counterfactual scenario?
> Conciseness: How clear and succinct is the text without unnecessary elaboration?
> Provide an overall score (out of 5) based on the combined evaluation of these aspects.

B Evaluation on Sentiment Analysis

To further verify the generalizability, we apply our method DICT to the Sentiment Analysis task and evaluate the performance on the SST-2 dataset. The details of SST-2 dataset are shown in Table 4. We compare our method DICT with RoBERTa-large (base model) and DISCO (the best baseline). The compared results (shown in Table 5) prove the effectiveness of our DICT on other NLP tasks. Our DICT can be generalized to various NLP tasks.

Data Split	Size
Train	67,350
Dev	873
Test	1,821

Table 3: Statistics of Dataset SST-2 for Sentiment Analysis.

Method	P	R	F1
RoBERTa-large	93.40	93.03	93.01
DISCO-14B	94.61	94.35	94.33
DICT-14B	**95.88**	**95.88**	**95.88**

Table 4: Performance comparison on Sentiment Analysis.

Method	Number of generated available counterfactual examples
AutoCAD	9,218
DISCO-7B	12,201
LLMCF-7B	12,033
LLMCF-14B	14,208
DICT-7B	16,012
DICT-14B	16,403

Table 5: Statistics of Generated CAD on SNLI-1.

Run	P	R	F1
1	65.17	65.21	64.89
2	65.28	65.16	64.92
3	64.84	64.88	64.85
Average	65.10	65.08	64.89

Table 6: Different Runs on SNLI-1.

C Statistics of Generated CAD

Taking the SNLI-1 dataset as an example, we perform counterfactual augmentation on each of the 20,000 samples across all methods. Notably, due to variations in the quality of counterfactual examples generated by different methods, the flip rate differs across them. As a result, the number of available counterfactual samples varies among the models. The details are provided in Table 5.

D Results of Different Runs on SNLI-1

We report the results of three runs of our DICT-14B on dataset SNLI-1 and show the mean results in Table 6. It shows that there is a slight fluctuation across different runs.