CAT: Causal Attention Tuning For Injecting Fine-grained Causal Knowledge into Large Language Models

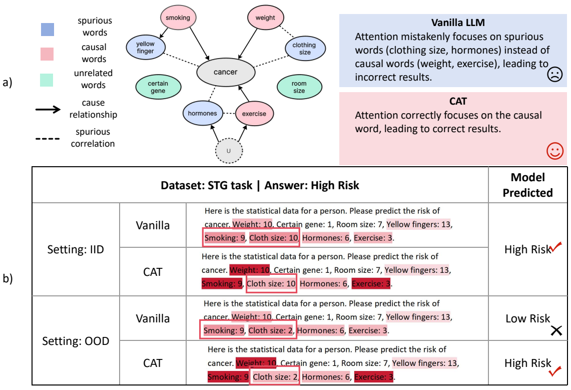
Kairong Han[1], Wenshuo Zhao[1], Ziyu Zhao[1], Junjian Ye[2], Lujia Pan[2], Kun Kuang[1†]
[1]College of Computer Science and Technology, Zhejiang University,
[2]Noah's Ark Lab, Huawei Technologies,
zju_handso@163.com, {zhao_ws, kunkuang}@zju.edu.cn,
benzhao.styx@gmail.com, {yejunjian, panlujia}@huawei.com

Abstract

Large Language Models (LLMs) have achieved remarkable success across various domains. However, a fundamental question remains: Can LLMs effectively utilize causal knowledge for prediction and generation? Through empirical studies, we find that LLMs trained directly on large-scale data often capture spurious correlations rather than true causal relationships, leading to suboptimal performance, especially in out-of-distribution (OOD) scenarios. To address this challenge, we propose Causal Attention Tuning (CAT), a novel approach that injects fine-grained causal knowledge into the attention mechanism. We propose an automated pipeline that leverages human priors to automatically generate token-level causal signals and introduce the Re-Attention mechanism to guide training, helping the model focus on causal structures while mitigating noise and biases in attention scores. Experimental results on our proposed Spurious Token Game (STG) benchmark and multiple downstream tasks demonstrate that our approach effectively leverages causal knowledge for prediction and remains robust in OOD scenarios. The CAT achieves an average improvement of 5.76% on the STG dataset and 1.56% on downstream tasks. Notably, the OOD performance of the Llama-3.1-8B model on STG_M increased from 64.5% to 90.5%, and Qwen's OOD performance on the STG_H dataset improved from 25.4% to 55.9%. Implementation details can be found here.

1 Introduction

Large Language Models (LLMs), trained in an autoregressive manner and guided by scaling laws, have achieved remarkable results across various domains (Zhao et al., 2023; Hadi et al., 2023; Zhou et al., 2024; Huang and Chang, 2022; Wang et al., 2024a; Hu et al., 2025a). Their foundational architecture, the Transformer (Vaswani, 2017), leverages the attention mechanism to capture token-level correlations, which is central to their success. However, existing fine-tuning paradigms primarily focus on aligning LLMs with task-specific objectives. Relying solely on superficial correlations in the data can lead to spurious correlations, causing biases and negatively impacting its reasoning ability and generalization (Wu et al., 2024a; Zhang et al., 2024a), which raises a critical question: *Can LLMs truly learn and utilize causal relationships, rather than merely modeling surface-level correlations?*



Vanilla LLM
Attention mistakenly focuses on spurious words (clothing size, hormones) instead of causal words (weight, exercise), leading to incorrect results. ☹

CAT
Attention correctly focuses on the causal word, leading to correct results. ☺

Dataset: STG task | Answer: High Risk			Model Predicted
Setting: IID	Vanilla	Here is the statistical data for a person. Please predict the risk of cancer. Weight: 10, Certain gene: 1, Room size: 7, Yellow fingers: 13, Smoking: 9, Cloth size: 10, Hormones: 6, Exercise: 3.	High Risk ✓
	CAT	Here is the statistical data for a person. Please predict the risk of cancer. Weight: 10, Certain gene: 1, Room size: 7, Yellow fingers: 13, Smoking: 9, Cloth size: 10, Hormones: 6, Exercise: 3.	
Setting: OOD	Vanilla	Here is the statistical data for a person. Please predict the risk of cancer. Weight: 10, Certain gene: 1, Room size: 7, Yellow fingers: 13, Smoking: 9, Cloth size: 2, Hormones: 6, Exercise: 3.	Low Risk ✗
	CAT	Here is the statistical data for a person. Please predict the risk of cancer. Weight: 10, Certain gene: 1, Room size: 7, Yellow fingers: 13, Smoking: 9, Cloth size: 2, Hormones: 6, Exercise: 3.	High Risk ✓

Figure 1: a) Training data is generated from this causal graph. LLM is influenced by spurious correlations and fails to learn causal relationships. b) The visualization of the attention distribution, where the deeper the red color, the higher the value. After training, the vanilla LLM incorrectly attends to spurious factors (e.g., clothing size), leading to failure in OOD scenarios. The CAT method, by injecting fine-grained causal knowledge, demonstrates stronger robustness in OOD scenarios.

To investigate this issue, we constructed a Spurious Token Game (STG) dataset, where the series attributes are classified into three types: causal factors, spurious factors(Ye et al., 2024b; Dong et al., 2025), and irrelevant factors. As illustrated in Figure 1, the numerical value of the spurious factor is proportional to the numerical value of the cor-

† Corresponding author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 9904–9921
November 4-9, 2025 ©2025 Association for Computational Linguistics

responding causal factor (i.e. cloth size has the same value as weight, 10 in the Figure). However, in the OOD (Liu et al., 2021; Tong et al., 2025, 2023) test, this association pattern is removed to verify whether the model can use causal knowledge for prediction (i.e., a change in clothing size to 2 does not affect cancer risk), more details in Appendix A. Our observations indicate that after direct fine-tuning, LLMs inherently allocate equal attention to both spurious and causal words, resulting in poor generalization in OOD scenarios (Dai et al., 2024; Gallegos et al., 2024). This suggests that direct fine-tuning often leads models to prioritize spurious correlations over genuine causal relationships, ultimately impairing their generalization ability. Consequently, this raises concerns about their robustness, as LLMs driven by associative learning tend to internalize dataset biases, making them less reliable in handling diverse and unpredictable real-world scenarios.

To solve this problem, we propose the **C**ausal **A**ttention **T**uning (CAT), a novel approach that integrates fine-grained causal knowledge into the attention mechanism. Specifically, the method consists of two steps. First, to automate the generation of causal supervision signals, human experts manually write a few examples and leverage an assistant LLM to generate causal supervision signals for a large-scale dataset. Second, to embed causal knowledge into attention, we convert word-level supervision signals into an adjacency matrix, aligning it with the attention training objective. Then, we introduce the Re-Attention mechanism, which guides model training by constraining the average attention map based on causal prior knowledge. Through the above approach, we align the decision-making process of LLMs with human causal knowledge at the attention level, effectively intervening in the model's decision dependencies. Finally, we develop a new benchmark STG to systematically evaluate whether LLMs can capture causal knowledge.

Experiments on STG demonstrate that CAT effectively directs the model's attention toward causal features, leading to consistent improvements in both independent and identically distributed (IID) and OOD settings. For example, the OOD performance of the Llama-3.1-8B model on STG_M increased from 64.5% to 90.5%, and Qwen's OOD performance on the STG_H dataset improved from 25.4% to 55.9%. Furthermore, evaluations on five widely used mathematical and reasoning tasks show that incorporating causal knowledge through CAT enhances the performance on downstream tasks. The CAT can be seamlessly integrated with mainstream training methods such as LoRA(Hu et al., 2021), demonstrating its general applicability. Our contributions are summarized as follows:

- We constructed a new benchmark called STG to evaluate whether LLMs can capture and use causal knowledge.
- We propose the CAT, a novel approach for integrating causal knowledge into LLMs. Through the Re-Attention mechanism, we mitigate noise and bias in the attention mechanism, resulting in performance improvements in both IID and OOD scenarios.
- Experiments on STG and mathematical and reasoning downstream tasks demonstrate the strong generalization ability of the CAT. The CAT achieves an average improvement of 5.76% on the STG dataset and 1.56% on downstream tasks. Notably, the OOD performance of the Llama-3.1-8B model on STG_M increased from 64.5% to 90.5%, and Qwen's OOD performance on the STG_H dataset improved from 25.4% to 55.9%, demonstrating the potential of the Re-Attention mechanism

2 Related Work

2.1 Combine Causality and LLMs

Since the advent of LLMs, researchers have explored ways to enhance their capabilities by integrating causal theory with LLMs (Wu et al., 2024a; Han et al., 2024).

In the areas of debiasing and fairness (Meade et al., 2021; Wang et al., 2024b), Counterfactual Data Augmentation (CDA) (Webster et al., 2020) is proposed to solve gender bias, which generates counterfactual samples by flipping gender-related keywords. Building on counterfactual generation, invariant loss (Zhou et al., 2023a) is introduced to further mitigate biases related to gender and other stereotypes. Entity bias (Longpre et al., 2021) is another kind of bias, and "do" operations (Pearl, 2010) on intermediate variables of both white-box and black-box large language models (Wang et al., 2023) is proposed to eliminate it. Jenny et al. employed activity dependency networks to better explain bias perspectives that were previously simplified only through correlations (Jenny et al., 2024). Zhou et al focused on conceptual bias and used counterfactual data generated by ChatGPT to balance label distributions and mitigate spurious correlations (Zhou et al., 2023b). Wu et al. propose the De-biased Attention Supervision (DAS) (Wu et al., 2024b) method, using the backdoor adjustment to mitigate bias caused by the label distribution of the dataset. However, the aforementioned debiasing works are limited to specific scenarios and have limited practical applicability. In terms of reasoning capabilities, causal prompt (Zhang et al., 2024a) is proposed, leveraging Chain-of-Thought (COT) for front-door adjustment to enhance reasoning performance, but the reasoning overhead is significant. Bao et al. and Li et al. addressed the issue of unfaithful COT by modeling and mitigating it from a causal explanation perspective (Bao et al., 2024; Li et al., 2024). Jin et al. introduced CausalCOT (Jin et al., 2023) to enhance causal reasoning abilities. Feng et al focused on learning a robust classifier across multiple domains (Feng et al., 2024).

Different from the aforementioned works, the CAT injects causal prior knowledge into the attention training process, offering a simple and efficient method with strong generalization ability for prediction and generation.

2.2 Research on Attention Mechanism

LLMs have sparked new explorations into the Attention mechanism. A series of studies focused on attention mechanisms (Niu et al., 2021; Guo et al., 2022; Hu et al., 2025b), aiming to explain and intervene in attention score distribution. Research has shown that the attention mechanism can extract reasonable word alignments, with attention scores and their norms collectively determining the output (Kobayashi et al., 2020). However, attention often allocates a significant portion of focus to tokens with no semantic value, a phenomenon termed "attention sinks" (Sun et al., 2024; Gu et al., 2024), which has been utilized to enhance long-context outputs (Xiao et al., 2023). However, maintaining these attention sinks is not always beneficial, and researchers have observed consistent performance improvements across various models by redistributing excess attention scores to other tokens (Yu et al., 2024). To optimize the attention distribution, the differential transformer (Ye et al., 2024a), inspired by signal denoising systems, adopts a sparse attention pattern, leading to performance improvements in LLMs. Nevertheless, the aforementioned methods overlook the inherent biases and spurious correlations in the data.

The CAT introduces causal prior supervision signals into attention training. By injecting fine-grained causal knowledge into the attention mechanism, we aim to accomplish debiasing and denoising at the architectural level, rather than merely fitting the data distribution of downstream tasks.

3 Preliminary

3.1 Attention Mechanism

We start with the vanilla Transformer, where an input sequence is mapped to a feature matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]^\top \in \mathbb{R}^{n \times d_{\text{model}}}$ through vocabulary embedding and positional encoding. Each of the n tokens is represented by a d_{model}-dimensional vector. We focus on the attention mechanism, which models dependencies between tokens:

$$\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i = \mathbf{S} \cdot \mathbf{W}_i^Q, \mathbf{S} \cdot \mathbf{W}_i^K, \mathbf{S} \cdot \mathbf{W}_i^V,$$

$$\mathbf{Z}_i^{attn} = \underbrace{\text{softmax}\left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_i^\top}{\sqrt{d_k}}\right)}_{\text{attention map}} \cdot \mathbf{V}_i,$$

$$\mathbf{Z}_{mult} = \text{Concat}(\mathbf{Z}_1^{attn}, \ldots, \mathbf{Z}_h^{attn}) \cdot \mathbf{W}_O,$$

where for each head i in multi-head attention, $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W_O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

3.2 Causal Graph

Causal graph is the structured representation of causal knowledge (Lipsky and Greenland, 2022; Thulasiraman and Swamy, 2011), denoted as a directed acyclic graph (DAG) $G = \{< V^G, E^G >\}$, where a directed edge $v_i \rightarrow v_j \in E^G$ indicates that element $v_i \in V^G$ is the direct cause of element $v_j \in V^G$, i.e., v_i causes v_j. We use the adjacency matrix corresponding to the causal graph DAG to align the training objective of attention.

4 Methodologies

In this section, we introduce the CAT framework, as shown in Figure 2. The CAT comprises two key steps: (1) causal prior knowledge extraction and (2) causal constraint attention training.

4.1 Causal Prior Knowledge Extraction

Due to the inherent complexity of natural language, aligning token-level causal relationships with existing causal prior knowledge presents three challenges:

- Causal relationships in natural language text are difficult to simply identify using rule-based matching.

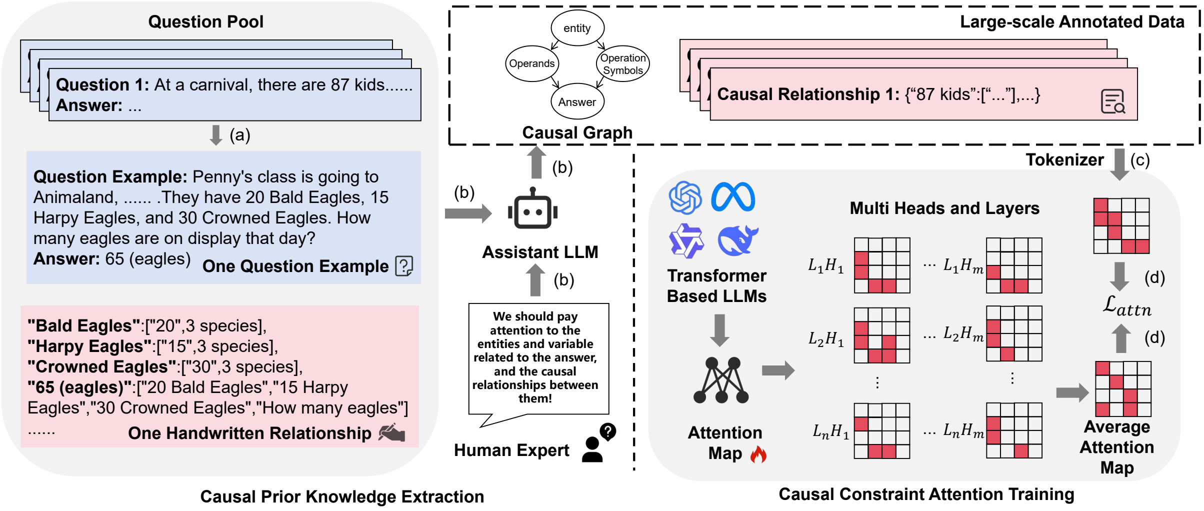

Question Pool
Question 1: At a carnival, there are 87 kids......
Answer: ...
(a)
Question Example: Penny's class is going to Animaland, ...They have 20 Bald Eagles, 15 Harpy Eagles, and 30 Crowned Eagles. How many eagles are on display that day?
Answer: 65 (eagles)
One Question Example
"Bald Eagles":["20",3 species],
"Harpy Eagles":["15",3 species],
"Crowned Eagles":["30",3 species],
"65 (eagles)":["20 Bald Eagles","15 Harpy Eagles","30 Crowned Eagles","How many eagles"]
......
One Handwritten Relationship
entity
Operands
Operation Symbols
Answer
Causal Graph
(b)
(b)
Assistant LLM
(b)
We should pay attention to the entities and variable related to the answer, and the causal relationships between them!
Human Expert
Large-scale Annotated Data
Causal Relationship 1: {"87 kids":["..."],...}
Tokenizer
(c)
Transformer Based LLMs
Attention Map
Multi Heads and Layers
L_1H_1
L_1H_m
L_2H_1
L_2H_m
L_nH_1
L_nH_m
(d)
$\mathcal{L}_{attn}$
(d)
Average Attention Map
Causal Prior Knowledge Extraction
Causal Constraint Attention Training

Figure 2: (a) Human experts construct handwritten causal relationships at the word level for downstream tasks. (b) The assistant LLM automates the annotation of downstream tasks based on handwritten examples. (c) Token-level causal associations are obtained using the tokenizer and transformed into an adjacency matrix. (d) The Re-Attention mechanism is employed to train LLMs by introducing $\mathcal{L}_{attn}$, which injects fine-grained parameterized causal knowledge to intervene the model's decision dependencies.

- The specific design of tokenizers can lead to the fragmentation of a single word into multiple tokens, adding complexity for LLMs in effectively incorporating causal knowledge.
- The high cost of large-scale annotation by human experts hinders scalability.

To address these challenges, we propose an automated pipeline for the generation of causal supervision signals.

Step 1: Prompt generation. Even though downstream tasks may involve complex and diverse expressions, the causal words that ultimately lead to the answer are considered to carry rich semantic information. In mathematical reasoning, we focus on numerical values, entities, numerical operation symbols, and the causal relationships between these words. Following this heuristic, we construct a prompt to guide the assistant LLM in uncovering causal relationships for the downstream task. The prompt consists of a task description $\mathcal{P}_t$ and handwritten specific examples $\mathcal{P}_d$, i.e., 65 eagles are calculated by 20 Bald, 15 Harpy, and 30 Crowned in Figure 2**(left)**. Although the causal graph is not explicitly provided, the handwritten examples follow the causal logic used by humans to solve the problem. The detailed prompt templates can be found in Appendix B.

Step 2: Token-Level causal knowledge extraction. Using large-scale human expert annotations is cost-intensive. so we annotate the training data by inputting $\mathcal{P}_t$ and $\mathcal{P}_d$ from Step 1, along with the downstream task question description $\mathcal{Q}$ and answer $\mathcal{A}$, into assistant LLM to obtain textual supervision signals $\mathcal{M}$:

$$\mathcal{M} = \text{LLM}([\mathcal{P}_t; \mathcal{P}_d; \mathcal{Q}; \mathcal{A}]).$$

To obtain a structured representation of causal relationships, we constrain the textual supervision signals to be in JSON format as a dictionary $\mathcal{M} = \{(k^M, v^M)|k^M \in \mathcal{Q} \bigcup \mathcal{A}, v^M \in \mathcal{Q} \bigcup \mathcal{A}\}$. This implies the generation of the k^M is primarily influenced by the v^M. Based on the specific tokenizer implementation, we convert the textual supervision signals into an adjacency matrix $\mathbf{A}^{adj} \in \{0,1\}^{n\times n}$:

$$\mathbf{A}^{adj}_{i,j} = \begin{cases} 1, & (token(i), token(j)) \in \mathcal{M} \\ 0, & \text{else} \end{cases}$$

where $\mathbf{A}^{adj}_{i,j} = 1$ indicates that the i-th token causes the j-th token, while $\mathbf{A}^{adj}_{i,j} = 0$ indicates no causal relationship. $(token(i), token(j))$ means the tuple formed by the words corresponding to the i-th and j-th tokens. Tokens identified as having causal relationships require additional attention during training.

4.2 Causal Constraint Attention Training

The attention map can be re-written in the following matrix form:

$$\mathbf{Z}_i^{attn} = \frac{1}{\sqrt{d_k}} \begin{bmatrix} a_{1,1} & 0 & \cdots & 0 \\ a_{2,1} & a_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix},$$

where $a_{i,j}$ denote the dot product $q_i \cdot k_j^\top$, and q_i is the query from the i-th token and k_j is the key from the j-th token. In next-token prediction, the i-th token serves as input for generating the $(i+1)$-th token. The model assigns attention weights to the first i tokens based on the i-th row of the attention map, where each $a_{i,j}$ reflects the importance of token j in predicting token $(i+1)$. To inject causal prior knowledge, we encourage the model to focus more on tokens that are causally related to the one being generated. This is achieved by shifting the token-level causal adjacency matrix $\mathbf{A}^{\text{adj}}$ upward by one position, aligning it with the next-token generation process:

$$\mathbf{A}_{i,j}^{adj} = \mathbf{A}_{i+1,j}^{adj}.$$

In multi-head attention, due to the difficulty in precisely quantifying the importance of different layers and different heads within each layer for downstream tasks, we consider the average attention map $\overline{\mathbf{A}^M}$ across all layers L and all heads H:

$$\overline{\mathbf{A}^M} = \frac{1}{L * H} \sum_{l=1}^{L} \sum_{h=1}^{H} \text{softmax}\left(\frac{\mathbf{Q}_{l,h} \cdot \mathbf{K}_{l,h}^\top}{\sqrt{d_k}} \right).$$

To enforce the attention mechanism to focus more on token-level causal relationships, we utilize the adjacency matrix of causal words from the previous section as a supervision signal. Specifically, in the average attention map $\overline{\mathbf{A}^M}$, for rows i where causal words appear, we calculate the average attention score $\mathcal{C}_i$ of the tokens corresponding to the causal words in that row. For the remaining tokens in the row, we compute the average attention score $\mathcal{N}_i$:

$$\mathcal{C}_i = \frac{1}{\sum_{j=1}^{i} \mathbf{A}_{i,j}^{adj}} \sum_{j=1}^{i} \overline{\mathbf{A}_{i,j}^M} \cdot \mathbf{A}_{i,j}^{adj},$$

$$\mathcal{N}_i = \frac{1}{\sum_{j=1}^{i} (1 - \mathbf{A}_{i,j}^{adj})} \sum_{j=1}^{i} \overline{\mathbf{A}^M}_{i,j} \cdot (1 - \mathbf{A}_{i,j}^{adj}).$$

We aim to ensure that the attention score of causal tokens in each row is no less than α times the average attention score of the remaining tokens. Therefore, we introduce the following loss:

$$\mathcal{L}_{attn} = \sum_{i=0}^{n} \max(0, \alpha - \frac{\mathcal{C}_i}{\mathcal{N}_i}).$$

This process allows attention to refocus on the causal relationship between tokens, the so-called ***Re-Attention*** mechanism, as shown in Figure 2(**right**). For both pre-training and supervised fine-tuning (SFT) of LLMs, researchers employ the next token prediction loss. Specifically, given a sequence of tokens $x_1, x_2, \ldots, x_T$ (Achiam et al., 2023; Devlin, 2018):

$$\mathcal{L}_{next} = - \sum_{t=1}^{T-1} \log P(x_{t+1} \mid x_{\leq t}).$$

Therefore, the total loss during training is:

$$\mathcal{L}_{total} = \mathcal{L}_{next} + \gamma \mathcal{L}_{attn},$$

where γ is used to modulate the gradient when applying constraints to the attention mechanism.

5 Experiments

5.1 Experimental Setup

Baseline. We conduct experiments on TinyLlama-1.1B[*] (Zhang et al., 2024b), Qwen2.5-1.5B[†] (Team, 2024; Yang et al., 2024) and Llama-3.1-8B-Instruct[‡] (Touvron et al., 2023), and conduct experiments under both full-parameter fine-tuning and parameter-efficient fine-tuning using LoRA(Hu et al., 2021).

Dataset. First, we evaluate CAT on two subsets of the STG dataset: STG_Easy (STG_E) and STG_Hard (STG_H). Furthermore, we used the following commonly used datasets related to mathematical reasoning, choice questions, and logical reasoning: MAWPS (Koncel-Kedziorski et al., 2016), ASDiv (Miao et al., 2020), GSM8K (Cobbe et al., 2021), ARC_E (Clark et al., 2018), and SVAMP (Patel et al., 2021). For the aforementioned data without a partitioned test set, we randomly split it into training, validation, and test sets with a ratio of 6:2:2.

Implementation Details. All experiments were conducted on the NVIDIA A100 40GB GPU. To guarantee the fairness of the comparison, we ensured that all hyperparameters were consistent between the baseline and the CAT. A warm-up coefficient of 0.1 was coupled with a cosine learning rate schedule and the AdamW optimizer. $\gamma = e^{-i}$ where i denotes the current epoch number. Unless

[*] https://huggingface.co/TinyLlama/TinyLlama_v1.1

[†] https://huggingface.co/Qwen/Qwen2.5-1.5B

[‡] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

otherwise specified, we use a default learning rate of $5 \times e^{-5}$ for full fine-tuning, and $1 \times e^{-4}$ for LoRA fine-tuning. The default number of training epochs is 4 for downstream tasks and 6 for the STG dataset. We use ChatGLM-4-air (GLM et al., 2024) as the assistant LLM. Other hyperparameter details and experiment details are shown in Appendix F.

5.2 Results

5.2.1 Spurious Tokens Game

The STG benchmark consists of two subsets: STG_E and STG_H. To investigate the impact of data volume on spurious correlations, STG_E is further divided into three scales of training datasets: large (STG_L), medium (STG_M), and small (STG_S), along with an IID testing set and an OOD testing set.

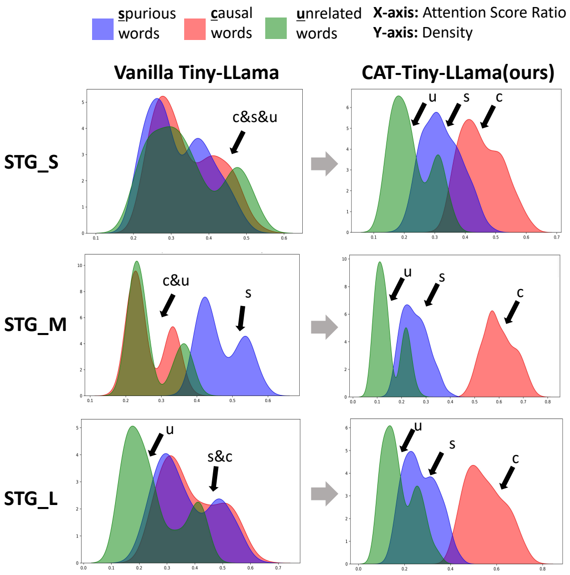


Figure 3: The density distribution of attention scores for three types of words in the average attention map under full parameters training when using TinyLlama-1.1B.

In STG_E, given a set of attributes and their corresponding values, the model learns to predict the risk of lung cancer. A specific example is illustrated in Figure 1. Cancer risks are only caused by causal factors $\mathcal{C}^s$. Spurious factors $\mathcal{S}^s$ have a proportional relationship with causal factors. Irrelevant factors $\mathcal{I}^s$ are sampled independently from the previous two. In OOD tasks, we break the proportional relationship between spurious association factors and causal factors. In STG_H, similar to STG_E, the model is required to make predictions based on input variables. However, the key difference is that STG_H includes a larger number of variables, and its answer is a continuous value ranging from 0 to 100, detailed in Appendix A. The experimental results are presented in Table 1.

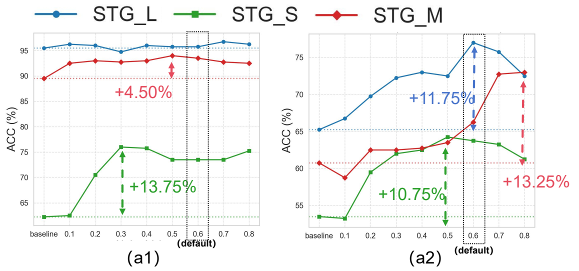


Figure 4: Results under full parameters training when using TinyLlama: a1) IID performance under different α, a2) OOD performance under different α.

The CAT achieves significant improvements over the baseline in both IID and OOD scenarios. We take TinyLlama-1.1B as an example to explore the impact details of CAT on the attention mechanism. We visualize the distribution function of the average attention scores for three types of factors, as shown in Figure 3. Additionally, we also analyze the impact of different α values on IID and OOD generalization, as shown in Figure 4.

Conclusion 1: The distribution of attention scores for different tokens, obtained spontaneously and unsupervised, is difficult to predict and unstable. This means unclear decision dependencies.

As shown in Figure 3, with small data scale, the baseline exhibits similar attention distributions across all factor types, resulting in near-random (50%) performance in both IID and OOD scenarios. With medium data, the baseline achieves 90% accuracy in IID but relies on spurious correlations, leading to poor OOD performance. With large data, the attention score is similarly distributed between causal and spurious correlation factors. Additionally, spurious correlations have introduced significant instability. With the Llama-3.1-8B model, when the data size is doubled (from _S to _M), although IID performance continues to improve, the OOD performance actually drops from 86.25% to 64.50%. Moreover, in STG_H, model performance consistently degrades greatly under OOD settings. In most settings, OOD performance is less than half of the IID performance. Simply scaling up the model size does not solve the problem.

As a comparison, with the CAT method, the improvement is significant. On average, CAT improves the IID performance by 3.95% and OOD performance by 7.56%. Notably, the OOD per-

Model	Setting	Task	Method	STG_E			STG_H	Average
				STG_S	STG_M	STG_L		
TinyLlama-1.1B	Full	IID	Vanilla	62.25%	89.50%	95.50%	32.20%	69.86%
			CAT	**73.50%**	**93.50%**	**95.75%**	**37.10%**	**74.96%**
		OOD	Vanilla	53.50%	60.75%	65.25%	4.10%	45.90%
			CAT	**63.75%**	**66.25%**	**77.00%**	**6.10%**	**53.27%**
	LoRA	IID	Vanilla	62.75%	83.50%	**96.00%**	31.90%	68.54%
			CAT	**81.50%**	**86.75%**	**96.00%**	**35.70%**	**74.99%**
		OOD	Vanilla	59.25%	56.75%	61.50%	5.90%	45.85%
			CAT	**65.50%**	**63.50%**	**69.50%**	**9.30%**	**51.95%**
Qwen2.5-1.5B	Full	IID	Vanilla	55.00%	**94.50%**	95.50%	56.60%	75.40%
			CAT	**74.00%**	**94.50%**	**95.75%**	**62.50%**	**81.69%**
		OOD	Vanilla	53.50%	**79.00%**	79.75%	25.40%	59.41%
			CAT	**62.50%**	**79.00%**	**83.25%**	**55.90%**	**70.16%**
	LoRA	IID	Vanilla	81.50%	**93.25%**	**95.75%**	51.50%	**80.50%**
			CAT	**82.00%**	90.50%	**95.75%**	**53.40%**	80.41%
		OOD	Vanilla	**78.50%**	82.00%	82.00%	38.70%	70.30%
			CAT	**78.50%**	**88.00%**	**84.25%**	**46.10%**	**74.21%**
Llama-3.1-8B	LoRA	IID	Vanilla	90.50%	93.25%	96.00%	57.80%	84.39%
			CAT	**94.00%**	**93.50%**	**96.75%**	**61.40%**	**86.41%**
		OOD	Vanilla	86.25%	64.50%	88.25%	49.60%	72.15%
			CAT	**89.00%**	**90.50%**	**89.25%**	**58.50%**	**81.81%**

Table 1: Comparison of IID and OOD experimental results between Vanilla and CAT on the STG task under different settings. Full represents full parameters training, LoRA represents LoRA training.

formance of the Llama-3.1-8B model on STG_M increased from 64.5% to 90.5%, and Qwen's OOD performance on the STG_H dataset improved from 25.40% to 55.9%.

Conclusion 2: As shown in Figure 4, within a certain range, the larger α, the better the performance.

As α increases, the model's attention to causal words grows. Across all dataset sizes, performance in both IID and OOD tasks initially improves. This suggests that greater attention to causal words enhances generalization. However, too much focus can disrupt the original attention distribution, causing a conflict with pre-trained parameters and resulting in performance degradation. More details in Appendix C.

Conclusion 3: The CAT enhances the model's performance by mitigating noise and bias.

As shown in Figure 5, without causal priors, the model fails in OOD scenarios, focusing on spurious correlation factors instead of causal factors. In contrast, CAT enables the model to focus on causal factors. The CAT changes the model's decision dependency mechanism, making its performance more robust.

5.2.2 Expand to Downstream Tasks

We evaluated a broader range of downstream tasks. Under the in-domain setting, we tested four datasets related to mathematics and reasoning. Furthermore, we introduce an out-of-domain setting: models trained on GSM8K are evaluated on other math reasoning datasets. Although these datasets involve basic arithmetic reasoning, they differ in question formulation and answer formats, making them distributionally distinct and suitable for assessing cross-task generalization. The results are shown in Table 2. The CAT has shown consistent improvements across multiple settings and datasets. For example, under the full fine-tuning setting with Qwen, our method yields an average performance improvement of 2.52%. Additionally, the CAT consistently outperforms the baseline in most OOD settings, demonstrating its stronger generalization ability. This suggests that guiding attention alignment toward human high-level causal reasoning

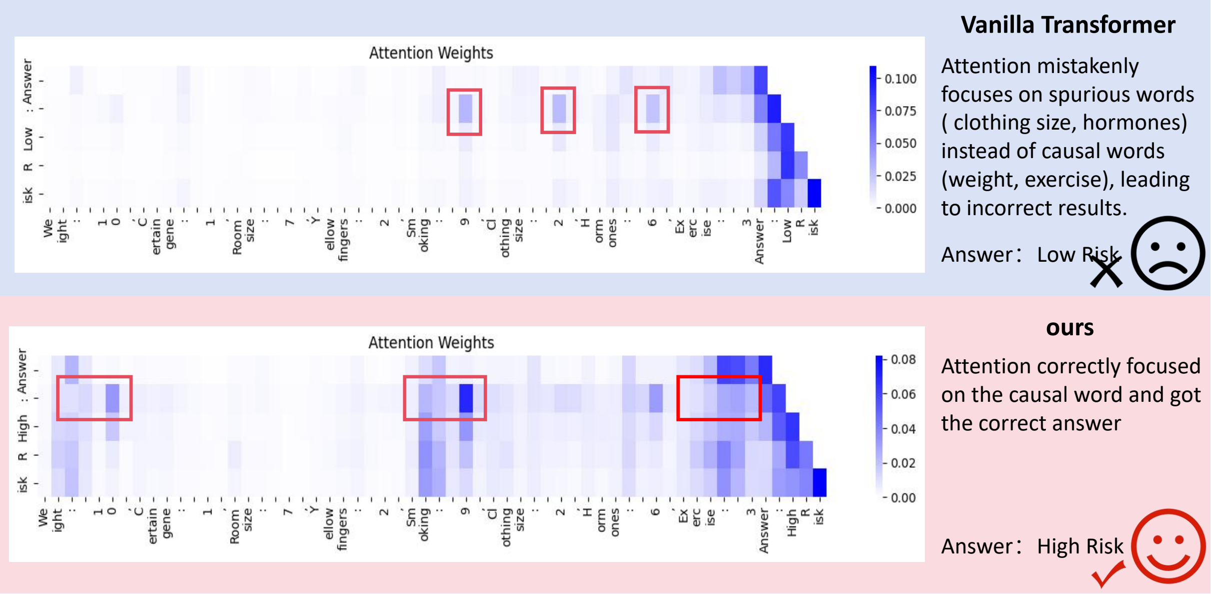


Figure 5: Visualization of the attention map in the STG task when using TinyLlama.

Model	Setting	Method	In-Domain				Out-of-Domain			Average
			MAWPS	SVAMP	ARC-E	GSM8K	ASDiv†	MAWPS†	SVAMP†	
TinyLlama-1.1B	Full	Vanilla	38.98%	17.50%	25.38%	13.04%	13.14%	20.68%	11.00%	19.96%
		CAT	**41.16%**	**20.00%**	**26.01%**	**14.18%**	**14.04%**	**23.15%**	**11.40%**	**21.42%**
		impv.	+2.18%	+2.50%	+0.63%	+1.14%	+0.90%	+2.47%	+0.40%	+1.46%
	LoRA	Vanilla	29.78%	10.50%	14.14%	**8.87%**	14.40%	**21.31%**	9.50%	15.50%
		CAT	**30.02%**	**12.00%**	**21.17%**	8.64%	**14.58%**	20.82%	**9.80%**	**16.72%**
		impv.	+0.24%	+1.50%	+7.03%	-0.23%	+0.18%	-0.49%	+0.30%	+1.22%
Qwen2.5-1.5B	Full	Vanilla	67.80%	51.00%	80.39%	45.34%	64.02%	79.52%	49.50%	62.51%
		CAT	**69.73%**	**56.00%**	**83.33%**	**47.08%**	**64.79%**	**82.18%**	**52.10%**	**65.03%**
		impv.	+1.93%	+5.00%	+2.94%	+1.74%	+0.77%	+2.66%	+2.60%	+2.52%
	LoRA	Vanilla	74.33%	64.50%	67.89%	47.23%	**70.52%**	89.88%	59.50%	67.69%
		CAT	**76.27%**	**65.00%**	**69.87%**	**50.04%**	68.58%	**90.02%**	**64.40%**	**69.17%**
		impv.	+1.94%	+0.50%	+1.98%	+2.81%	-1.94%	+0.14%	+4.90%	+1.48%
Llama-3.1-8B	LoRA	Vanilla	89.83%	72.00%	91.58%	65.66%	76.66%	90.94%	66.20%	78.98%
		CAT	**90.31%**	**72.50%**	**91.84%**	**66.57%**	**78.51%**	**91.33%**	**69.70%**	**80.11%**
		impv.	+0.48%	+0.50%	+0.26%	+0.91%	+1.85%	+0.39%	+3.50%	+1.13%

Table 2: Performance comparison for different models and tasks. Full represents full parameters training. LoRA represents LoRA training."†" means training on GSM8K but testing on the given dataset.

can help models acquire deeper reasoning capabilities, rather than simply fitting to the training distribution.

5.2.3 Ablation Studies

To explore the influence of the α and the γ, we conducted ablation experiments, taking Qwen2.5-1.5B as an example. We train LoRA and set α to 0.05, 0.1, 015, 0.2, 0.25, and 0.3. For γ, we set the coefficient γ of $\mathcal{L}_{attn}$ as 1 (w/o γ). Using a weight decay strategy proves beneficial in most cases, as shown in Table 3. As α increases, the performance gradually improves, shown in Figure 6. Through ablation experiments, we have demonstrated the effectiveness of each component of the CAT.

Mehod	MAWPS	SVAMP	ARC-E	GSM8K	Average
CAT	69.73%	**56.00%**	**83.33%**	**47.08%**	**64.03%**
w/o γ	**71.91%**	54.50%	82.58%	45.64%	63.66%

Table 3: Ablation experiment on γ.

5.2.4 Cost Analysis and Powerful Assistant LLMs

We replace ChatGLM-4-air with GPT-4o as the assistant LLMs, details in Appendix E. Stronger assistant LLMs exhibit slightly better performance. Additionally, when using ChatGLM-4-air, the an-

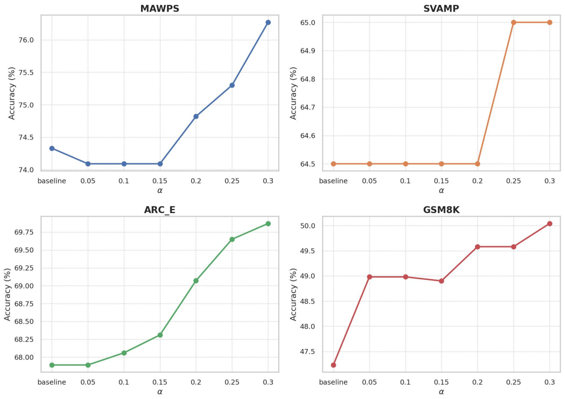


Figure 6: Performance under different α using LoRA.

notation cost can be as low as \$0.14[§] per million tokens, in contrast to approximately \$18 with GPT-4o. Further details are provided in the Appendix D. Therefore, considering the cost, we recommend using GLM-4-air for better cost-effectiveness.

6 Conclusions

To address the issues of spurious correlations and the lack of causal knowledge in the inherent correlation-based Transformer, we propose CAT, a novel method that injects fine-grained causal knowledge during training. To evaluate the IID and OOD robustness, we introduce the STG benchmark. Extensive experiments across downstream tasks under diverse settings validate the positive impact of incorporating fine-grained causal knowledge and the *Re-Attention* mechanism in downstream tasks. LLMs can effectively utilize causal knowledge for prediction and generation.

Limitations

Due to resource limitations, we did not explore the performance impact of larger models, such as those exceeding 10B parameters (i.e., Qwen-2.5 14B), under the CAT method. Experimenting with larger LLMs could provide stronger insights. Additionally, CAT requires the introduction of an assistant LLM to label causal supervision signals, which will incur extra token overhead, although these costs remain within an acceptable range. This paper offers an empirical approach to causal knowledge injection. However, there is still significant room for exploration in terms of how to integrate causal knowledge into LLM mechanisms, starting from a more theoretical token-level causal modeling perspective. Due to the complexity of causal theory, even well-intentioned and fully professional humans may cause an LLM assistant to inject biases that do not exist in the training data. Causal relationships in the real world may be more complex, abstract, and context-dependent. The applicability of our approach to tasks that require a deeper and more nuanced understanding of causality has not yet been fully explored.

Ethical Concerns

We declare that all authors of this paper acknowledge the *ACM Code of Ethics* and honor the code of conduct. We do not foresee an immediate ethical or societal impact resulting from our work. However, our method provides opportunities for human experts to maliciously inject biases into LLMs, for example, by downplaying the causal effects of belonging to socially marginalized groups or by exaggerating the apparent correlation of spurious factors. Therefore, we urge users to exercise caution when using this method to avoid potential ethical and moral risks.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China (2024YFE0203700), National Natural Science Foundation of China (62376243), "Pioneer" and "Leading Goose" R&D Program of Zhejiang (2025C02037), and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-0010). All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Guangsheng Bao, Hongbo Zhang, Linyi Yang, Cunxiang Wang, and Yue Zhang. 2024. Llms with chain-of-thought are non-causal reasoners. *arXiv preprint arXiv:2402.16048*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind

[§] The pricing unit of the GLM API is in CNY, approximately 1 CNY per million tokens, which is equivalent to about 0.14 USD based on the exchange rate as of August 26, 2025.

Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. Bias and unfairness in information retrieval systems: New challenges in the llm era. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6437–6447.

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

Xinpeng Dong, Min Zhang, Didi Zhu, Ye Jun Jian, Zhang Keli, Aimin Zhou, Fei Wu, and Kun Kuang. 2025. Erict: Enhancing robustness by identifying concept tokens in zero-shot vision language models. In *Forty-second International Conference on Machine Learning*.

Tao Feng, Lizhen Qu, Zhuang Li, Haolan Zhan, Yuncheng Hua, and Reza Haf. 2024. IMO: Greedy layer-wise sparse representation learning for out-of-distribution text classification with pre-trained models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2625–2639, Bangkok, Thailand. Association for Computational Linguistics.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Meng-Hao Guo, Tian-Xing Xu, Jiang-Jiang Liu, Zheng-Ning Liu, Peng-Tao Jiang, Tai-Jiang Mu, Song-Hai Zhang, Ralph R Martin, Ming-Ming Cheng, and Shi-Min Hu. 2022. Attention mechanisms in computer vision: A survey. *Computational visual media*, 8(3):331–368.

Muhammad Usman Hadi, Rizwan Qureshi, Abbas Shah, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, Seyedali Mirjalili, et al. 2023. A survey on large language models: Applications, challenges, limitations, and practical usage. *Authorea Preprints*.

Kairong Han, Kun Kuang, Ziyu Zhao, Junjian Ye, and Fei Wu. 2024. Causal agent based on large language model. *arXiv preprint arXiv:2408.06849*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Zijing Hu, Fengda Zhang, Long Chen, Kun Kuang, Jiahui Li, Kaifeng Gao, Jun Xiao, Xin Wang, and Wenwu Zhu. 2025a. Towards better alignment: Training diffusion models with reinforcement learning against sparse rewards. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23604–23614.

Zijing Hu, Fengda Zhang, and Kun Kuang. 2025b. D-fusion: Direct preference optimization for aligning diffusion models with visually consistent samples. *Preprint*, arXiv:2505.22002.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

David Jenny, Yann Billeter, Bernhard Schölkopf, and Zhijing Jin. 2024. Exploring the jungle of bias: Political bias attribution in language models via dependency analysis. In *Proceedings of the Third Workshop on NLP for Positive Impact*, pages 152–178.

Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, LYU Zhiheng, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, et al. 2023. Cladder: Assessing causal reasoning in language models. In *Thirty-seventh conference on neural information processing systems*.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Jiachun Li, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. 2024. Towards faithful chain-of-thought: Large language models are bridging reasoners. *arXiv preprint arXiv:2405.18915*.

Ari M Lipsky and Sander Greenland. 2022. Causal directed acyclic graphs. *JAMA*, 327(11):1083–1084.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. 2021. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Nicholas Meade, Elinor Poole-Dayan, and Siva Reddy. 2021. An empirical survey of the effectiveness of debiasing techniques for pre-trained language models. *arXiv preprint arXiv:2110.08527*.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984.

Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. 2021. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Judea Pearl. 2010. Causal inference. *Causality: objectives and assessment*, pages 39–58.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

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

Krishnaiyan Thulasiraman and Madisetti NS Swamy. 2011. *Graphs: theory and algorithms*. John Wiley & Sons.

Yunze Tong, Junkun Yuan, Min Zhang, Didi Zhu, Keli Zhang, Fei Wu, and Kun Kuang. 2023. Quantitatively measuring and contrastively exploring heterogeneity for domain generalization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Yunze Tong, Fengda Zhang, Zihao Tang, Kaifeng Gao, Kai Huang, Pengfei Lyu, Jun Xiao, and Kun Kuang. 2025. Latent score-based reweighting for robust classification on imbalanced tabular data. In *Proceedings of the 42nd International Conference on Machine Learning*.

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

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, Singapore. Association for Computational Linguistics.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Xiyao Wang, Yuhang Zhou, Xiaoyu Liu, Hongjin Lu, Yuancheng Xu, Feihong He, Jaehong Yoon, Taixi Lu, Gedas Bertasius, Mohit Bansal, et al. 2024b. Mementos: A comprehensive benchmark for multimodal large language model reasoning over image sequences. *arXiv preprint arXiv:2401.10529*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Anpeng Wu, Kun Kuang, Minqin Zhu, Yingrong Wang, Yujia Zheng, Kairong Han, Baohong Li, Guangyi Chen, Fei Wu, and Kun Zhang. 2024a. Causality for large language models. *arXiv preprint arXiv:2410.15319*.

Yiquan Wu, Yifei Liu, Ziyu Zhao, Weiming Lu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2024b. De-biased attention supervision for text classification with causality. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19279–19287.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

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.*

Tianzhu Ye, Li Dong, Yuqing Xia, Yutao Sun, Yi Zhu, Gao Huang, and Furu Wei. 2024a. Differential transformer. *arXiv preprint arXiv:2410.05258.*

Wenqian Ye, Guangtao Zheng, Xu Cao, Yunsheng Ma, and Aidong Zhang. 2024b. Spurious correlations in machine learning: A survey. *arXiv preprint arXiv:2402.12715.*

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan (Celine) Lin. 2024. Unveiling and harnessing hidden attention sinks: enhancing large language models without training through attention calibration. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Congzhi Zhang, Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. 2024a. Causal prompting: Debiasing large language model prompting based on front-door adjustment. *arXiv preprint arXiv:2403.02738.*

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. 2024b. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385.*

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

Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2024. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, pages 1–65.

Fan Zhou, Yuzhou Mao, Liu Yu, Yi Yang, and Ting Zhong. 2023a. Causal-debias: Unifying debiasing in pretrained language models and fine-tuning via causal invariant learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4227–4241.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2023b. Explore spurious correlations at the concept level in language models for text classification. *arXiv preprint arXiv:2311.08648.*

A Data Generate Process

Machine learning theory posits that training and test sets are IID. However, due to the presence of spurious correlation, although the model's outcomes should be uniquely determined by causal features, the model may inadvertently capture these spurious correlations. This can lead to a reliance on spurious correlations rather than causal features when faced with a wide and diverse array of real-world scenarios, thereby compromising the model's reliability. So, for STG_E, our data generation processing follows the formalized expression below, where in the IID scenario, it satisfies:

$$\mathcal{C}_i^s, \mathcal{I}_i^s \sim Rand(1, 10)$$

$$\mathcal{S}_i^s = r_i * \mathcal{C}_i^s$$

$$f(\mathcal{C}^s) = \sum_i k_i * \mathcal{C}_i^s$$

$$\mathcal{A}^s = \begin{cases} High, & f(\mathcal{C}^s) \geq \mu_h \\ Low, & else \end{cases}$$

where r_i, k_i, μ_h are hyperparameters to control the ratio of high risk and low risk. The accuracy of random guessing is 50%.

In the OOD scenario, the three elements are independent of each other:

$$\mathcal{S}_i^{ood}, \mathcal{C}_i^{ood}, \mathcal{I}_i^{ood} \sim Rand(1, 10)$$

A specific example is as follows:

> **Question:** Here is the statistical data for a person. Please predict the probability of cancer.
> Yellow fingers: 3, Weight: 1, Room size: 4, Certain gene: 4, Clothing size: 1, Smoking: 2, Hormones: 2, Exercise: 5 Here is the statistical data for a person. Please predict the probability of cancer.
> **Answer:** Low Risk

Specifically, the value of yellow fingers is 1.5 times that of smoking, the value of clothing size is the same as weight, and the value of hormones is 0.5 times that of exercise. All values are rounded down. $\mu_h = 7.2$ and

$$f(\mathcal{C}^s) = 1.2 * \#Smoking + 0.7 * \#Weight - \#Exercise$$

For the STG_H dataset, we follow the causal graph shown in Figure 7. The process is similar to that of STG_E, and the implementation details can be found in our code repository. The specific differences among the subsets of the STG dataset are shown in Table 4.

Name	Subset	Training Size	Test Size	OOD Test Size	Node Number	Answer
STG_E	STG_S	0.4k	0.4k	0.4k	8	high/low risk
STG_E	STG_M	0.8k	0.4k	0.4k	8	high/low risk
STG_E	STG_L	1.6k	0.4k	0.4k	8	high/low risk
STG_H	–	3k	1k	1k	14	0-100

Table 4: Details of STG dataset subsets.

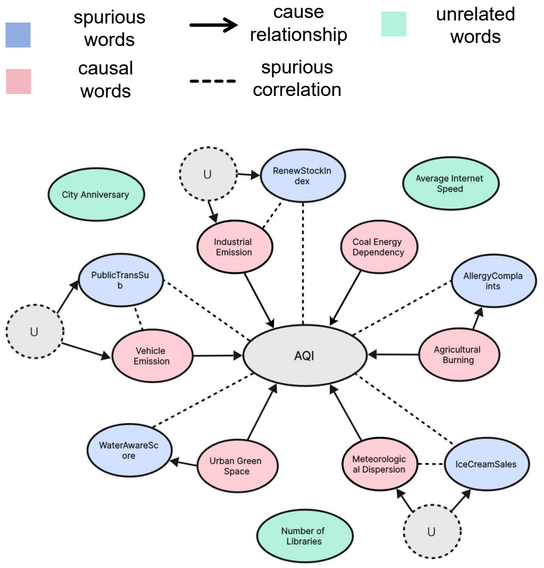


Figure 7: The causal graph underlying the data generation process of the STG_H dataset.

B Prompt Template

In different downstream tasks, we manually crafted prompt templates to guide the assistant LLM in extracting token-level causal associations. The prompt templates for different tasks are as follows:

SVAMP

You need to evaluate the causal importance relationships between tokens in text data from the field of mathematical reasoning. Among them, entities, values, and keywords containing operation symbols are crucial for numerical reasoning. The data is used to train autoregressive models, so tokens that appear later can only see the tokens that come before them. Please output the important tokens for executing mathematical reasoning tasks during training, along with the tokens they should focus on from the preceding context as causal associations (which can be more than one). Present the output JSON string in a dict format, such as {"A":[...], "B":[...],...}. You should only output JSON without other contents. Note that the Answer part is considered important and must be analyzed.

##demo

If they are already at 659 feet and the cave is 762 feet deep. How much farther until they reach the end of the cave?**Answer:** 103.0

##output

{
"762 feet deep":["the cave"],
"until":["How much farther"],
"Answer":["659 feet","762 feet", "until", "end of the cave"],
"103.0":["659 feet", "and", "762 feet", "Answer"]
}

##Please output following sentence importance between tokens. The final answer at the end and the corresponding number's importance must always be analyzed (such as 103.0 shown above).

ARC_E

You need to evaluate the causal importance relationships between tokens in text data from the field of reasoning. You only need to consider the tokens that have the greatest impact on the final answer. The data is used to train autoregressive models, so tokens that appear later can only see the tokens that come before them. Please output the important tokens for executing reasoning tasks during training, along with the tokens they should focus on from the preceding context as causal associations (which can be more than one). Present the output JSON string in a dict format, such as {"A":[...],"B":[...],...}. Note that the Answer part is considered important and must be analyzed. Below I will give you a single-choice question. You need to analyze the most important part of each option for the answer, and together with the answer, form the causal relationship that needs to be considered to generate the answer. Note that only the token behind can notice the previous word, and keep the autoregressive characteristics, such as "option content": "option A/B/C/D". The specific example is as follows:

##demo:

Which factor will most likely cause a person to develop a fever?

A. a leg muscle relaxing after exercise
B. a bacterial population in the bloodstream
C. several viral particles on the skin
D. carbohydrates being digested in the stomach
Answer: B

##output:

{
"develop a fever":["factor","cause"],
"leg muscle relaxing":["A."],
"bacterial population":["B."],
"viral particles":["C."],
"digested in the stomach":["D."],
"Answer: B":["A.", "leg muscle relaxing", "B.","bacterial population", "C.","viral particles", "D.", "digested in the stomach"]
}

##Please output following sentence importance between tokens. The final answer at the end and the corresponding number's importance must always be analyzed (such as Answer: B shown above). You should only output JSON string without other contents.

GSM8k

You need to evaluate the causal importance relationships between tokens in text data from the field of mathematical reasoning. Among them, entities, values, and keywords containing operation symbols are crucial for numerical reasoning. The data is used to train autoregressive models, so tokens that appear later can only see the tokens that come before them. Please output the important tokens for executing mathematical reasoning tasks during training, along with the tokens they should focus on from the preceding context as causal associations (which can be more than one). Present the output JSON string in a dict format, such as {"A":[...],"B":[...],...}. You should only output JSON without other contents. Note that the Answer part is considered important and must be analyzed.

##demo

Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May? Answer: Natalia sold 482 = «482=24»24 clips in May. Natalia sold 48+24 = «48+24=72»72 clips altogether in April and May.#### 72

##output

{
"in April":["48"],
"in May": ["half as many clips", "482 = «482=24»24 clips", "48"],
"72 clips": ["How many clips", "sell altogether", "48+24", "in April", "in May"],
"#### 72":["How many clips","in April and May","48+24","72 clips"]
}

##Please output following sentence importance between tokens. The final answer at the end and the corresponding number's importance must always be analyzed (such as #### 72 shown above). Please try to use the most refined causal characteristics to summarize the causal process of the answer

MAWPS

You need to evaluate the causal importance relationships between tokens in text data from the field of mathematical reasoning. Among them, entities, values, and keywords containing operation symbols are crucial for numerical reasoning. The data is used to train autoregressive models, so tokens that appear later can only see the tokens that come before them. Please output the important tokens for executing mathematical reasoning tasks during training, along with the tokens they should focus on from the preceding context as causal associations (which can be more than one). Present the output JSON string in a dict format, such as "A":[...],"B":[...],.... You should only output JSON without other contents. Note that the Answer part is considered important and must be analyzed.

##demo

William has 2 bottle caps. He buys 41 more. How many bottle caps does William have in all? Answer: 43.0

##output

{
"2 bottle caps": ["William"],
"41 more": ["He buys"],
"William have": ["How many bottle caps"],
"Answer": ["How many bottle caps"],
"43.0": ["2 bottle caps", "41 more"]
}

##Please output following sentence importance between tokens. The final answer at the end and the corresponding number's importance must always be analyzed (such as 43.0 shown above).

C Details of the Experimental Results on the STG Dataset

We visualized the distribution of attention scores for STG under different α under TinyLlama-1.1B, as shown in Figure 9, 11, 10.

As we can see, as α increases, the model's attention to causal words gradually strengthens, and within a certain range, both IID and OOD performance improve consistently.

In addition, we analyzed the performance changes of LoRA setting under different α, as shown in Figure 8.

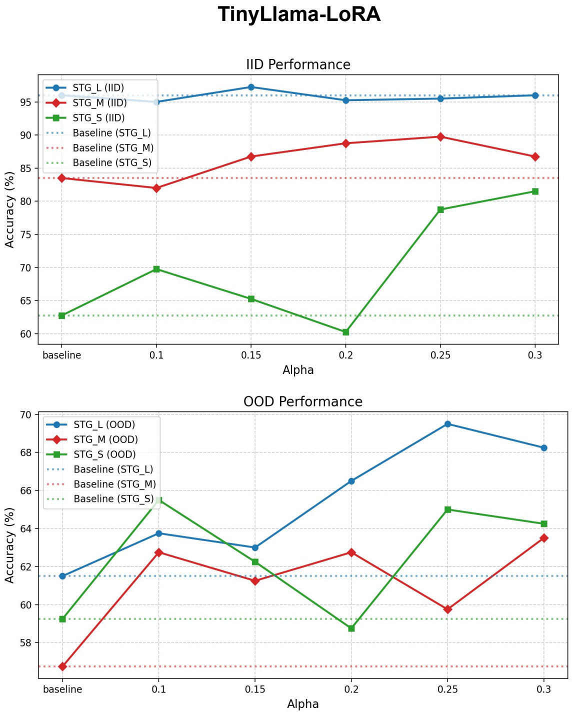


Figure 8: Performance of STG dataset under different α using TinyLlama-1.1B-LoRA.

D Cost Analysis

To estimate annotation costs, we randomly sampled 10 instances from the GSM8K dataset and annotated them using GPT-4o. The statistics for these samples are as follows: the average token length of the original inputs was 168.4, the average length of the prompts (including instructions for causal supervision signal extraction) was 570.0, and the average length of the model's output completions was 163.9.

Based on the official pricing of GPT-4o, we estimate the maximum additional cost per 1 million tokens annotated, assuming no cache hits, as:

$$\left(\frac{570.0}{168.4} \times \$2.50\right) + \left(\frac{163.9}{168.4} \times \$10.00\right) \approx \$18.19$$

For comparison, we also consider the use of ChatGLM-4-air, which was employed in our experiments. Given that the output lengths across different LLMs are relatively consistent, we adopt the same average prompt and output lengths as a reasonable approximation. Based on the official pricing of ChatGLM-4-air, the estimated cost is:

$$\frac{570.0 + 163.9}{168.4} \times 0.25 \approx 1.09(CNY)$$

These results demonstrate the low cost of our approach. By utilizing batch API calls to proprietary large language models, our method enables efficient large-scale data annotation. Moreover, it is compatible with mainstream closed-source models (e.g., GPT-4, ChatGLM, Gemini), making it adaptable to various application requirements.

E Comparison of Different Assistant LLMs

To fairly compare the performance of different assistant LLMs. We use GPT4o to replace GLM-4-air and conduct comparative experiments in

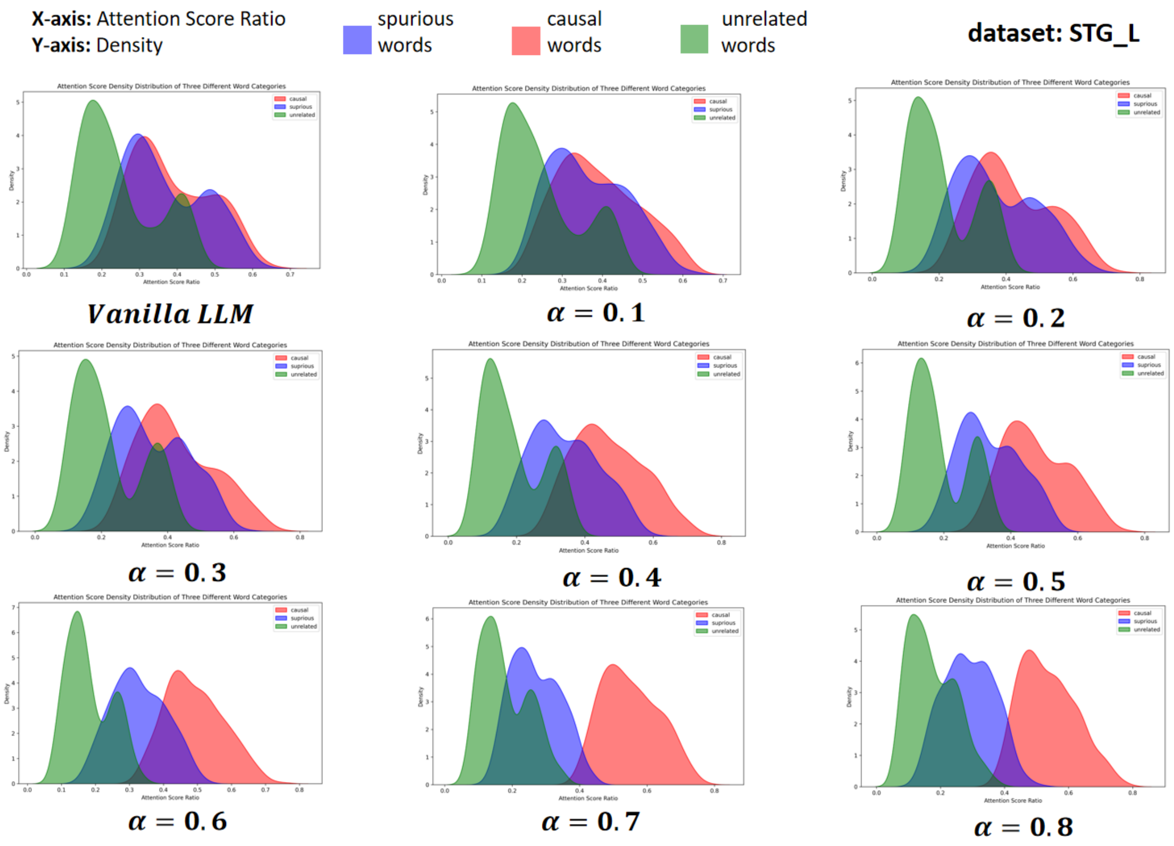


Figure 9: Visualization of the attention distribution on the STG_L dataset.

MAWPS. Specifically, we conducted six experiments on the α grid of 0.05, 0.1, 0.15, 0.2, 0.25, and 0.3, and reported the average and max accuracy. The experimental results are shown in Table 5. Stronger teacher models exhibit slightly better performance. Considering the cost, we recommend using ChatGLM-4-air.

Model	Setting	Source	Avg ± Std	Max
Tinyllama-1.1B	Full	GPT	0.4165 ± 0.0143	0.4310
		GLM	**0.4185 ± 0.0216**	**0.4552**
	Lora	GPT	**0.2986 ± 0.0023**	**0.3027**
		GLM	0.2982 ± 0.0026	0.3002
Qwen2.5-1.5B	Full	GPT	**0.7288 ± 0.0157**	**0.7482**
		GLM-4	0.7123 ± 0.0099	0.7264
	Lora	GPT	**0.7502 ± 0.0055**	0.7579
		GLM	0.7478 ± 0.0081	**0.7627**
Llama-3.1-8B	Lora	GPT	0.8979 ± 0.0055	**0.9055**
		GLM-4	**0.9003 ± 0.0047**	**0.9055**

Table 5: The impact of different assistant LLMs on performance. GPT represents GPT-4o, and GLM represents ChatGLM-4-air.

F Details of the Hyperparameters Used During Training.

All comparisons between baselines and the CAT are obtained after sufficient training with the same hyperparameter settings. Due to differences in tasks, models, and GPU memory limitations, we ensure that the product of batch size and gradient accumulation steps remains consistent across different downstream tasks under the same model setting. For testing, we use greedy decoding with the following parameter settings, as shown in Table 6.

Parameter	Value
max_new_tokens	512
batchsize	64
num_return_sequences	1
do_sample	False

Table 6: Hyperparameters used during testing

All random seeds used in this paper are set to 42 to ensure the reproducibility of the experiments. For more experimental details, please refer to our code repository.

Due to differences in attention distributions

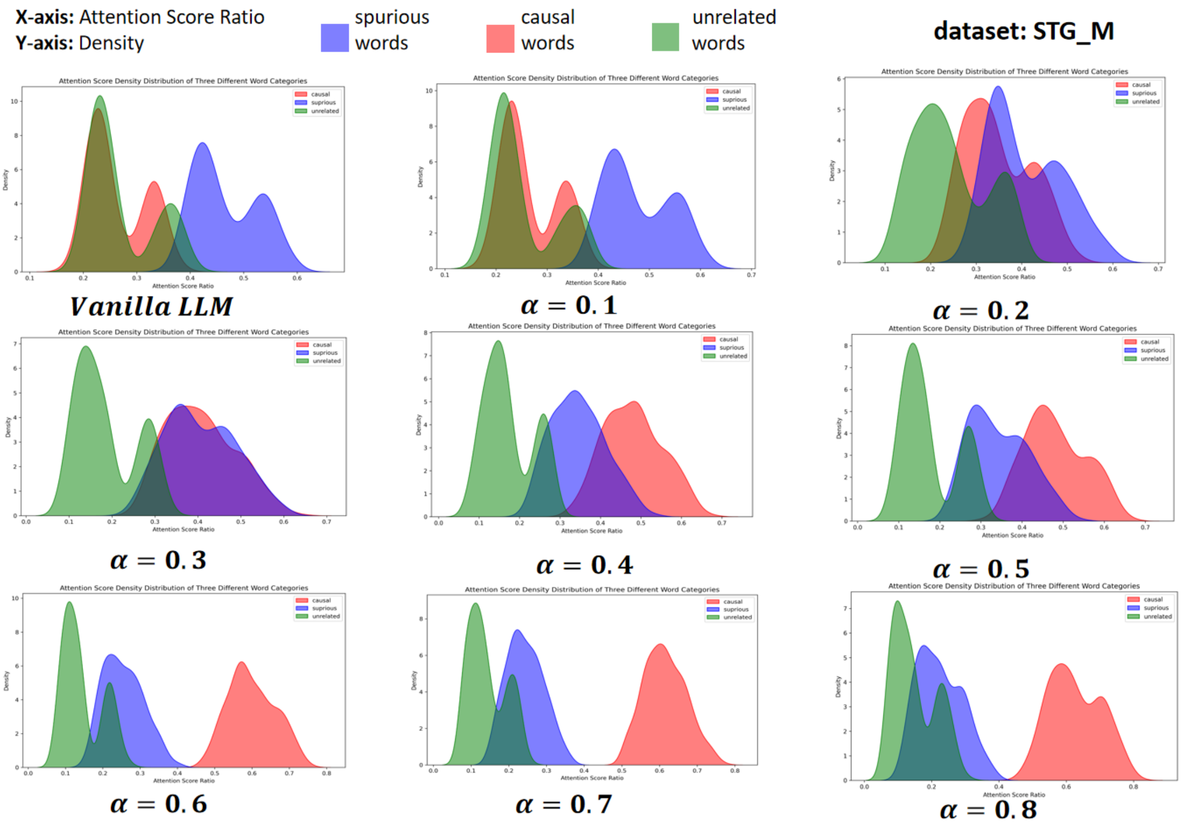


Figure 10: Visualization of the attention distribution on the STG_M dataset.

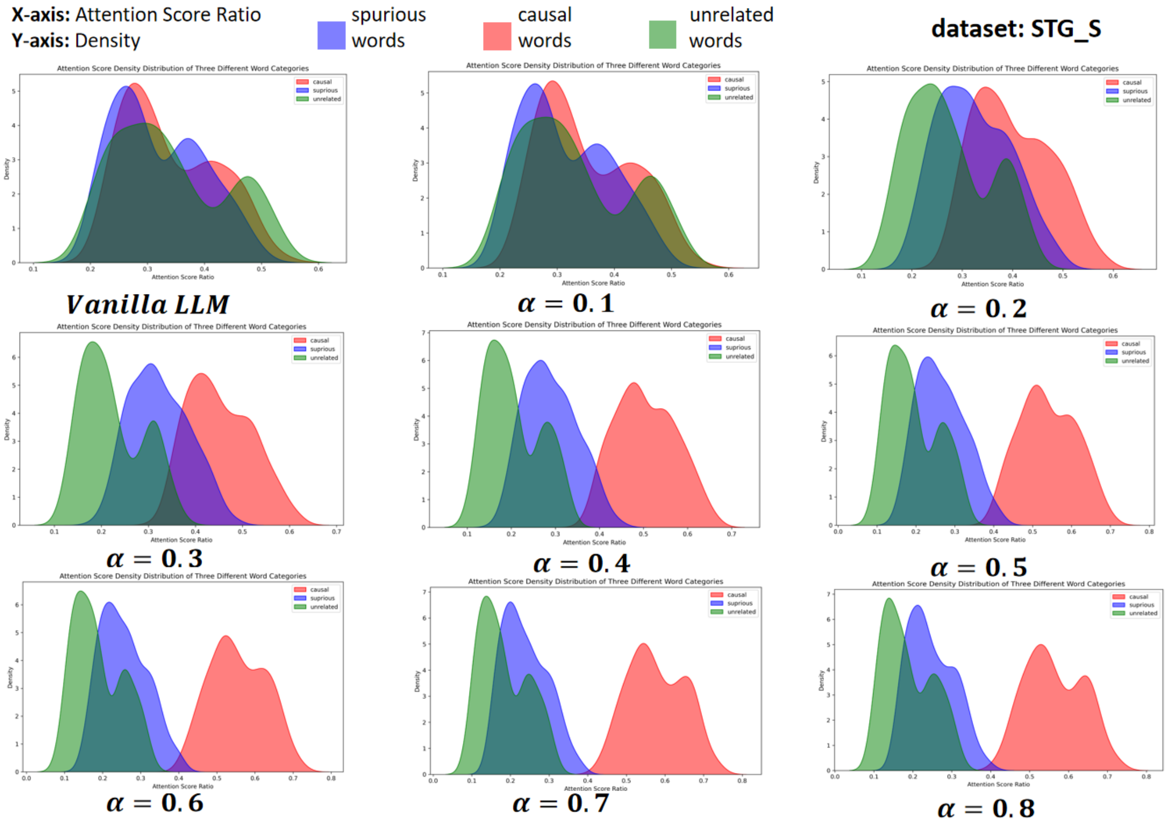


Figure 11: Visualization of the attention distribution on the STG_S dataset.

across different models and tasks, we observe empirically that setting the α parameter between 0.05 and 0.35 yields better results. We strive to keep the α hyperparameter consistent within the same model whenever possible. In the five reasoning downstream tasks, we report the test results based on the best-performing model on the validation set. For TinyLlama-1.1B LoRA, we set α to 0.2. For Qwen-2.5-1.5B, both LoRA and the full model, we use 0.3. For TinyLlama-1.1B, we use two settings: 0.15 and 0.2. For Llama-3.1-8B-Instruct, we use 0.25 and 0.3. For STG_H, we uniformly set the α parameter to 0.3 for all LoRA models and 0.2 for all full-parameter models. For STG_E, we perform a grid search with an interval of 0.05 in the range from 0.05 to 0.35. Specifically, for the TinyLlama-1.1B full-parameter model on STG_E, we set α to 0.6. The α parameter determines the degree to which the model relies on causal associations. How to efficiently identify the optimal α value is left for future work.

For the STG_E dataset, we apply LoRA fine-tuning to TinyLlama and Qwen using a learning rate of $1 \times e^{-3}$, and full fine-tuning on Qwen with a learning rate of $1 \times e^{-4}$.

G Use Of AI Assistants

We used generative AI, ChatGPT, to check for syntactic and grammatical errors in the manuscript. We carefully verified the correctness of the revised content.