

ExpCSD: Experience-based Debiasing Framework for Conversational Stance Detection Grounded in Latent Thinking Paths

Xiang Zeng¹, Ye Wang², Xin Song¹, Dong Zhu¹, Wenqing Hou¹, Bin Zhou^{1,*}

¹National University of Defense Technology, Changsha, China

²Harbin Institute of Technology, Shenzhen, China

{zengxiang, wangye, songxin, zhudong, houwenqing, binzhou}@nudt.edu.cn

Abstract

Conversational stance detection, which aims to identify users attitudes in conversation threads, has attracted increasing attention in recent years. Yet, mainstream LLM-based approaches remain prone to biases caused by spurious correlations. Existing debiasing methods either rely on indiscriminate multi-step prompting, which can induce unnecessary reasoning and overthinking, or require expensive fine-tuning on curated data. To address these challenges, we propose ExpCSD, an experience-based debiasing framework grounded in latent thinking paths that follows a debias-only-when-needed principle. ExpCSD first assesses whether an LLMs latent thinking path is biased and selectively retrieves and injects historically similar debiasing experiences only when necessary, without updating model parameters. Experiments on public benchmarks show that ExpCSD outperforms strong baselines in stance detection accuracy while avoiding overthinking and costly fine-tuning.

1 Introduction

Conversational stance detection, which identifies users' attitudes toward a target in multi-turn dialogues (Niu et al., 2024b; Wang et al., 2025a), leverages conversational context to provide a more comprehensive analysis than single-sentence methods (Li et al., 2021) and has attracted increasing attention.

With their strong contextual understanding, large language models (LLMs) play an important role in conversational stance detection. However, LLMs are susceptible to biases caused by spurious correlations (Li et al., 2025a), which undermines the reliability of stance detection. As shown in Figure.1, User 3's negative sentiment is directed at User 2's way of speaking rather than at the target SpaceX, yet the model still predicts a biased

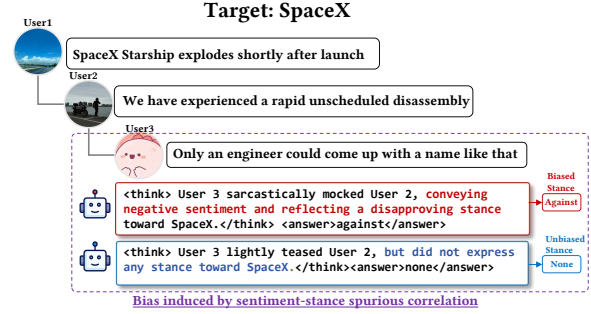


Figure 1: Bias example in conversational stance detection.

against stance toward SpaceX simply because of this negative affect, revealing a sentimentstance spurious correlation.

Existing debiasing efforts for stance detection with LLMs largely fall into two paradigms. (1) **Prompt-engineering approaches** reduce bias by inducing more comprehensive reasoning, e.g., contrasting rationales from different perspectives (Weinzierl and Harabagiu, 2024; Zhang et al., 2025b) or multi-agent debate (Lan et al., 2024; Wang et al., 2024a). While effective in some cases, they often apply multi-step deliberation to every sample, which can unnecessarily amplify uncertain evidence and then shift inherently neutral samples toward non-neutral labels (an overthinking failure mode (Dang et al., 2025)). (2) **Fine-tuning approaches** (Li et al., 2025b; Zou et al., 2025; Wang et al., 2025a) mitigate bias by constructing bias-reduced data and updating LLM parameters, but they are constrained by the limited scale of curated unbiased data and the increasing cost of scaling up fine-tuning.

To avoid overthinking induced by indiscriminate intervention, we explore whether potential biased reasoning can be detected before debiasing. Following mechanistic studies of LLMs (Ye et al., 2025), we regard the progressive hidden states as observable proxies for internal information pro-

*Corresponding author

cessing and refer to them as *latent thinking paths* for brevity. The intuition is that comprehensive stance reasoning integrates targets, context, and sentiment relations, leading to richer representational transitions. In contrast, shortcut reasoning often relies on a small set of superficial cues, such as sentiment polarity, and therefore tends to involve fewer state updates and simpler latent paths.

We quantify latent thinking paths using geometric features, including magnitude variation and angular shifts. Our comparative analysis shows that biased samples tend to exhibit stronger angular deviations but lower exploration magnitude in the latent space. Furthermore, counterfactual experiments show that weakening shortcut cues reduces biased-path scores, suggesting a close association between geometric features and spurious-cue dependency, as detailed in Section 4.4.1. Recent studies also report that biased or incorrect reasoning is accompanied by distinctive magnitude and angular patterns in hidden representations (Wang et al., 2025b; Du et al., 2025), suggesting that flawed reasoning may correspond to simpler latent paths and less stable semantic modeling. These geometric patterns support online detection of potential shortcut reasoning and motivate our on-demand debiasing paradigm. Specifically, latent paths serve as indicators for applying debiasing only to potentially biased cases, thereby alleviating redundant reasoning and reducing computational cost.

Therefore, we propose **ExpCSD**, an **Experience-based framework for Conversational Stance Detection** grounded in latent thinking paths. Following a debias-only-when-needed principle, ExpCSD first assesses whether a thinking path is biased and selectively injects historically similar debiasing experiences to revise reasoning without fine-tuning. ExpCSD consists of four components: (1) **Thinking path modeling**. We represent the progression of hidden states as a latent thinking path and encode it with geometric features that capture magnitude variations and angular shifts. We use prediction correctness as a weak supervision signal to construct candidate biased and unbiased path libraries. (2) **Experience generation**. To correct bias, we start from biased samples and compare its biased reasoning with correct reasoning to get debiasing experience that can generalize. But *typicality bias* (Zhang et al., 2025a) can cause the model to repeatedly produce biased reasoning; therefore, we adopt

stance verbalized sampling to elicit correct reasoning and improve experience generation. (3) **Experience filtering**. To remove ineffective experience, we use a two-stage filter. It keeps only experiences that (i) correct biased stances and (ii) greatly improve the unbiasedness score of the debiased thinking path. (4) **Inference-time debiasing**. At inference time, we follow the “debias-only-when-needed” principle. The model first makes an initial prediction and its thinking-path embedding. Only if the path is closer to the biased library than to the unbiased library, we retrieve similar debiasing experiences to debias the prediction. This gated procedure mitigates bias while avoiding unnecessary deliberation, and experiments on public benchmarks show that ExpCSD outperforms strong baselines without costly fine-tuning.

Our contributions are threefold: (1) We propose ExpCSD, a debias-only-when-needed experience-based framework that detects biased latent thinking paths and selectively injects similar debiasing experiences without updating LLM parameters. (2) We introduce stance verbalized sampling to elicit diverse reasoning rationales, facilitating effective experience generation, and a debiasing utility score to filter out ineffective experiences. (3) Experiments on two public benchmarks show that ExpCSD achieves state-of-the-art overall-F1.

2 Related Work

2.1 Conversational Stance Detection

Conversational stance detection (CSD) aims to identify a users attitudes toward a target in multi-turn dialogues. (Li et al., 2023b) first formalized the task and released a dataset, while (Niu et al., 2024b) constructed the MT-CSD benchmark and leveraged pre-trained language models to capture global and local dialogue context. More recently, LLM-based methods (Wang et al., 2025a; Niu et al., 2024a) have become the dominant paradigm due to strong contextual understanding, with approaches like (Wang et al., 2025a) using multi-agent reasoning to enhance fine-grained semantic and logical reasoning. Despite these advances, LLM-based methods remain susceptible to inherent model biases, undermining CSD reliability.

2.2 Mitigating Biases in Stance Detection

Mitigating biases in LLM-based stance detection can be grouped into two paradigms. (1)

Prompting-based methods guide models to generate and compare reasoning rationales from multiple perspectives or adopt multi-agent debates. For example, ToC (Weinzierl and Harabagiu, 2024) constructs and compares counterfactual rationales across stances for a single sample; MPVS-tance (Zhang et al., 2025b) seeks stance evidence from multiple perspectives; COLD (Lan et al., 2024) and DEEM (Wang et al., 2024a) employ role-based multi-agent discussions. (2) Fine-tuning methods update model parameters with relatively unbiased data. FACTUAL (Li et al., 2025a) constructs counterfactual datasets to debias LLMs via a calibration network; TAT (Zou et al., 2025) reconstructs reasoning processes grounded in gold labels for dual-objective fine-tuning; MRGKD (Wang et al., 2025a) leverages multi-agent reasoning graphs for fine-tuning. Despite improvements, this paradigm is constrained by limited unbiased data, leaving residual bias and limiting overall debiasing effectiveness.

3 Methodology

3.1 Task Definition

Conversational stance detection aims to identify the stance expressed by users in social media conversations toward a specific target. Given the target $\mathcal{T}$, the current user’s utterance u_c , and n -turn conversation history $U = \{u_1, u_2, \dots, u_n\}$, where each u_i is a contextual comment and u_{i+1} replies to u_i , the goal is to leverage U to enhance the understanding of the current utterance u_c , thereby determining the stance of the user towards the target $\mathcal{T}$: $\hat{s} = M_\theta(\mathcal{T}, u_c, U)$, where M_θ represents the stance detection model, and $\hat{s}$ represents the predicted stance.

3.2 Model Overview

As shown in Fig.2, ExpCSD comprises four modules, with their interrelations detailed in the figure caption.

3.2.1 Thinking Path Modeling

To determine whether the current thinking path is biased, we first need to model the feature of the thinking path. Inspired by (Ye et al., 2025; Wang et al., 2025b), we treat the progressive hidden states as the latent thinking path of LLMs. Specifically, assume the model f has L hidden layers; we decompose f into ordered sub-modules:

$$f = f_{\text{emb}} \circ f_1 \circ \dots \circ f_l \circ \dots \circ f_L \circ f_{\text{head}}. \quad (1)$$

where $f_{\text{emb}} : \mathbb{R}^{|\mathcal{V}|} \rightarrow \mathbb{R}^d$ (denoted as f_0) is the 0-th embedding layer, $f_{\text{head}} : \mathbb{R}^d \rightarrow \mathbb{R}^{|\mathcal{V}|}$ is the final classification layer, and f_l ($1 \leq l \leq L$) : $\mathbb{R}^d \rightarrow \mathbb{R}^d$ are intermediate hidden layers. Here d is the embedding dimension and $\mathcal{V}$ is the vocabulary.

First, the target $\mathcal{T}$, comment u_c , and conversation history U are converted into the model f input text x using the native stance detection prompt p_{naive} , as shown in Fig.12, and model f outputs $\hat{y} = \hat{y}_1 \hat{y}_2 \dots \hat{y}_T$ (T tokens). For the t -th token, its l -th layer hidden state is z_l^t . Following the definitions in (Wang et al., 2024b, 2025b), we define the average embedding at layer l as $h_l = \frac{1}{T} \sum_{t=1}^T z_l^t$, which represents the l -th sentence hidden state. The latent Thinking Path Embedding Chain (TPEC) of LLM is a progressive chain H of all sentence hidden states: $H = h_0 \rightarrow \dots h_l \dots \rightarrow h_L$.

To capture the thinking path feature, its geometric information (Rintoul and Wilson, 2015) is the most fundamental dimension, which usually includes magnitude $M(h_t, h_{t+1})$ and angle $A(h_t, h_{t+1})$ that can reflect the distance and direction changes produced during the path wandering. The two measures are formalized as follows :

$$\begin{aligned} M(h_t, h_{t+1}) &= \|h_{t+1} - h_t\|_2, \\ A(h_t, h_{t+1}) &= \arccos \left(\frac{h_{t+1}^\top h_t}{\|h_{t+1}\|_2 \cdot \|h_t\|_2} \right). \end{aligned} \quad (2)$$

To better fuse the two features of magnitude and angle for a more accurate characterization of the thinking path, and following the definitions in (Wang et al., 2025b), we introduce the complex plane to combine $M(h_t, h_{t+1})$ and $A(h_t, h_{t+1})$ into a new feature point $C(h_t, h_{t+1})$, which can take into account both the change angle and magnitude of thinking states evolution from the l -th layer to the $(l+1)$ -th layer. Here, $M(h_t, h_{t+1})$ represents the complex magnitude and $A(h_t, h_{t+1})$ represents the complex argument. This feature point $C(h_t, h_{t+1})$ can be expressed as:

$$\begin{aligned} C(h_l, h_{l+1}) &= M(h_l, h_{l+1}) \cdot e^{i \cdot A(h_l, h_{l+1})} \\ &= \underbrace{M(h_l, h_{l+1}) \cos A_l}_{\text{Re}(C(h_l, h_{l+1}))} + i \underbrace{M(h_l, h_{l+1}) \sin A_l}_{\text{Im}(C(h_l, h_{l+1}))}, \end{aligned} \quad (3)$$

where $A_l = A(h_l, h_{l+1})$. We construct a complex vector c by concatenating all layer-wise complex feature points $C(h_l, h_{l+1})$ (for $l = 0, 1, \dots, L-1$) in the order of TPEC to characterize thinking path

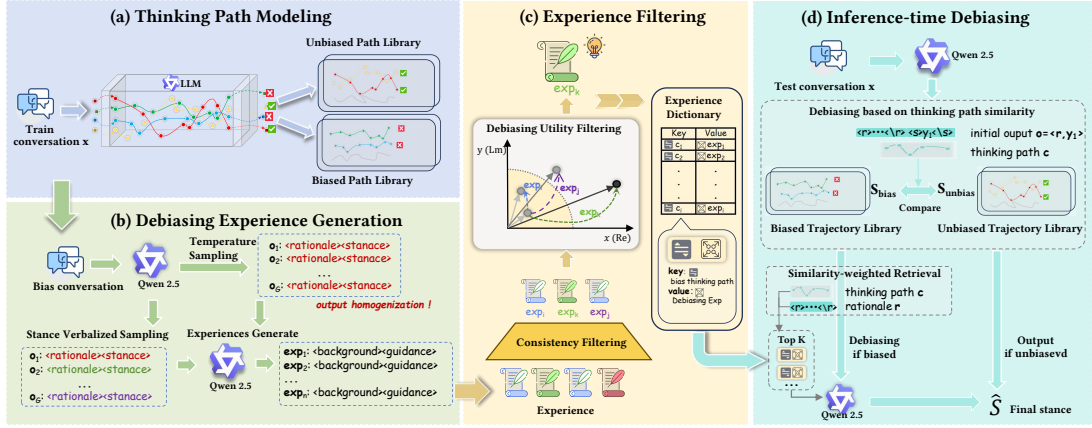


Figure 2: The proposed framework for Exp-CSD. It consists of four core modules: (a) **Thinking Path Modeling:** Model LLMs latent thinking paths and, using prediction correctness as supervision, partition training instances into biased and unbiased thinking-path libraries; (b) **Debiasing Experience Generation:** Compare diverse reasoning traces from biased instances to identify the causes of bias and generate generalizable debiasing experiences; (c) **Experience Filtering:** Apply a two-stage filtering mechanism to retain only effective experiences in the experience dictionary; (d) **Inference-time Debiasing:** Apply selective debiasing by retrieving relevant debiasing experiences only for predictions whose thinking paths are more similar to the biased library.

layer-by-layer evolution:

$$c = \left[C(h_0, h_1), \dots, C(h_{L-1}, h_L) \right]^T \in \mathbb{C}^L \quad (4)$$

After obtaining the thinking-path features, we follow prior conversational stance detection studies (Zou et al., 2025; Wang et al., 2025a) in constructing unbiased data by using training labels as supervision. Specifically, we partition the thinking paths of the training set into an unbiased library $D_{\text{unbias}} = \{ \langle x_i, c_i \rangle \mid \hat{y}_i = y_i, \forall i \in \{1, 2, \dots, N\} \}$ when predictions match the ground-truth labels, and a biased library $D_{\text{bias}} = \{ \langle x_i, c_i \rangle \mid \hat{y}_i \neq y_i, \forall i \in \{1, 2, \dots, N\} \}$ otherwise. where y_i denotes the ground-truth stance label corresponding to x_i .

3.2.2 Debiasing Experience Generation

Inspired by the training-free GRPO paradigm (Cai et al., 2025), which extracts optimization experience by comparing multiple rollouts for the same incorrect instance, as shown in Fig.2(b), we adopt a similar strategy to automatically generate debiasing experiences from biased samples. Specifically, given a biased instance x_i sampled from the biased thinking path library D_{bias} with ground-truth label y_i , we perform G independent rollouts with the LLM, obtaining a set of outputs $\mathcal{O}_i = \{ o_i^{(1)}, o_i^{(2)}, \dots, o_i^{(G)} \}$. We then provide $\mathcal{O}_i$ along with y_i to an experience-generation prompt

p_{exp} and query the LLM to compare reasoning traces that lead to correct versus incorrect outcomes (w.r.t. y_i). From this comparative analysis, the LLM distills a set of debiasing experiences $\mathcal{E}_i = \{ E_i^{(j)} = \langle b_i^{(j)}, g_i^{(j)} \rangle \mid j = 1, \dots, n \}$. Here, $b_i^{(j)}$ describes the biased reasoning scenario where the experience applies, and $g_i^{(j)}$ provides the corresponding generalizable debiasing guidance. However, extracting debiasing experiences from biased data is challenging due to typicality bias (Zhang et al., 2025a): even with high-temperature sampling, the model repeatedly produces homogeneous biased outputs, yielding little informative contrast between correct and incorrect traces despite label supervision. To mitigate this, we propose *stance verbalized sampling* (SVS), which uses a prompt p_{svs} to encourage sampling reasoning paths with different stances from low-probability regions, thereby eliciting correct and diverse trajectories. Contrasting these trajectories under label supervision helps the model understand both *what* the correct stance is and *why* it is correct, providing more effective guidance for steering the model toward the correct stance and enabling effective, generalizable debiasing experiences. The above process of extracting debiasing experiences via contrastive output is formally defined as follows:

$$\begin{cases} \mathcal{O}_i^{\text{naive}} = LLM(p_{\text{naive}}, x), \\ \mathcal{O}_i^{\text{svs}} = LLM(p_{\text{svs}}, x), \\ \mathcal{E}_i = LLM(p_{\text{exp}}, \mathcal{O}_i^{\text{naive}}, \mathcal{O}_i^{\text{svs}}, y_i) \end{cases} \quad (5)$$

Here, O_i^{naive} and O_i^{svs} are reasoning outputs generated under prompt p_{naive} and p_{svs} , respectively. The p_{exp} is the experience generate prompt. The full details of the p_{exp} and p_{svs} are as Fig.7 and Fig.8 (Appendix §A).

3.2.3 Experience Filtering

Although stance verbalized sampling enriches debiasing experience generation, it still yields ineffective experiences. To address this issue, we propose a two-stage filtering mechanism to preserve only high-quality, effective debiasing experiences.

Tier I: Consistency-based Filtering. We first keep only experiences that can correct biased predictions, i.e., guiding the model to produce outputs consistent with the ground-truth labels. Experiences that fail to achieve basic correction are discarded. Formally, an consistent experience set $\mathcal{E}_I = \{E_j \mid \hat{y}_i^{E_j} = y_i, E_j \in \mathcal{E}_I\}$, where $\hat{y}_i^{E_j}, \hat{H}_i^{E_j} = \text{LLM}(p_{\text{debiasing}}, x_i, E_j)$ denotes the outputs of LLMs after incorporating the experience E_j into the debiasing prompt $p_{\text{debiasing}}$ (See Fig.9 for the full prompt), $\hat{H}_i^{E_j}$ represents the debiased reasoning path, $\hat{y}_i^{E_j}$ is the prediction result for the biased sample x_i , and y_i denotes the corresponding ground-truth label.

Tier II: Debiasing Utility Filtering. After consistency filtering, we select experiences whose inclusion maximally increases the unbiasedness of the thinking path. Analysis of angles and magnitudes (As analyzed in section 4.4.1) shows that unbiased paths have larger magnitudes and smaller angles. Therefore, large magnitude and small angle can serve as a measure of a paths unbiasedness. To integrate magnitude and angle, each pair of adjacent states is mapped onto a complex plane as a feature point $C(h_l, h_{l+1})$ (Eq.3). For a thinking path H , the real and imaginary parts of all feature points are summed and averaged, and the modulus of the resulting vector is used as the debiasing utility. The debiasing utility score $U(H^{E_j})$ for E_j is calculated as follows:

$$U(H^{E_j}) = \sqrt{\left(\frac{1}{L} \sum_{l=0}^{L-1} \text{Re}(C(h_l, h_{l+1}))\right)^2 + \left(\frac{1}{L} \sum_{l=0}^{L-1} \text{Im}(C(h_l, h_{l+1}))\right)^2} \quad (6)$$

$U(H^{E_j})$ captures the joint effects of magnitude and angle, yielding larger values for correct paths with aligned vectors and lower values for biased paths due to vector cancellation caused by larger angular deviations. In short, this metric quantifies how much each experience con-

tributes to debiasing a thinking path. Finally, let $\mathcal{D}_{\text{exp}}$ denote the experience dictionary. As shown in Fig. 2(c), we select the experience E^* with the highest debiasing-utility score and store it in the dictionary: $\mathcal{D}_{\text{exp}}[c] = E^*$, where $E^* = \arg \max_{E_j \in \mathcal{E}_I} U(H^{E_j})$. where the dictionary key consists of the biased thinking path feature vector c , and the value is the debiasing experience.

3.2.4 Inference-time Debiasing

After constructing a high-quality experience dictionary, we perform debiasing during the inference stage. As shown in Fig.2(d), given a test sample x , we first apply the naive prompt p_{naive} to obtain an initial prediction $o = \langle r, y_1 \rangle$, where r denotes the reasoning rationale and y_1 denotes the predicted stance. We then extract the corresponding latent thinking path embedding c . Next, we assess whether the current thinking path is biased by comparing its similarity to a biased path library $\mathcal{D}_{\text{bias}}$ and an unbiased path library $\mathcal{D}_{\text{unbias}}$. Let c_i^{bias} and c_i^{unbias} denote the i -th thinking path embeddings in the biased and unbiased libraries, respectively. The similarity scores are computed as the maximum cosine similarity within each library: $S_{\text{bias}} = \max_{i \in \mathcal{D}_{\text{bias}}} \cos(c, c_i^{\text{bias}})$, $S_{\text{unbias}} = \max_{i \in \mathcal{D}_{\text{unbias}}} \cos(c, c_i^{\text{unbias}})$. If $S_{\text{bias}} > S_{\text{unbias}}$, the prediction is considered potentially biased, and the debiasing process is triggered. During debiasing, we retrieve the top- K experiences from the dictionary $\mathcal{D}_{\text{exp}}$ based on a weighted similarity between the thinking path c and experience background b :

$$S(E_j) = \alpha \cos(c, c_j) + (1 - \alpha) \cos(\text{Enc}(r), \text{Enc}(b_j)). \quad (7)$$

where $\text{Enc}(\cdot)$ is a SBERT encoder and $\alpha \in [0, 1]$ balances path and background similarity. These experiences are injected into $p_{\text{debiasing}}$ to obtain y_2 , with the final stance set to y_2 if bias is detected, otherwise y_1 .

4 Experiments

4.1 Experimental Setups

Datasets: We evaluate MRGKD on two public stance detection datasets: SRQ (Villa-Cox et al., 2020), a Twitter dataset with four targets and MT-CSD (Niu et al., 2024b), a conversational benchmark covering five topics. For more details, see Appendix §B.1.

Evaluation metrics: We adopt F1-avg as the evaluation metric to evaluate the performance of

Method	Bitcoin	Tesla	SpaceX	Biden	Trump	Overall
BERT	53.60	47.39	49.31	29.13	45.11	44.91
TTS	53.60	46.08	52.41	31.23	44.41	45.55
GLAN	56.95	52.38	55.98	38.15	<u>48.91</u>	50.47
GPT-3.5-turbo	46.89	51.69	53.16	36.05	27.47	43.05
GPT-4	49.39	50.71	55.34	45.09	40.33	48.17
DeepSeek	47.25	47.53	55.68	35.48	40.11	45.21
Doubao-pro	53.76	55.56	55.18	35.59	47.73	49.56
COLA	54.55	53.48	56.19	40.73	43.09	49.61
ToC	56.06	55.61	54.32	41.24	42.33	49.31
MPVStance	57.29	54.35	51.69	42.74	45.25	50.26
FACTUAL	50.21	49.94	53.01	37.24	46.52	47.38
MRGKD	<u>59.25</u>	<u>56.71</u>	<u>56.73</u>	43.70	44.90	<u>52.26</u>
ExpCSD (ours)	59.77	60.37	61.33	44.80	51.29	55.51

Table 1: Results (%) on MT-CSD dataset in in-target setting. Best scores are in **bold**. The suboptimal scores are underlined. The significance tests of our ExpCSD over MRGKD and GPT4 (with $p < 0.05$).

stance detection methods, consistent with the approaches in (Li et al., 2021) and (Niu et al., 2024b). F1-avg represents the average F1 score computed for the “against” and “favor” stances.

Baselines: We compare four categories of methods: (1) **Fine-tuning PLMs:** BERT (Devlin et al., 2019), TTS (Li et al., 2023a) (target-based data augmentation), and GLAN (Niu et al., 2024b) (GCN-CNN structural modeling); (2) **In-context Learning with LLMs:** GPT-3.5 Turbo, GPT-4, Doubao-Pro, and DeepSeek-Chat (each with one demonstration); (3) **Prompt Engineering with LLMs:** COLA (Lan et al., 2024) (role-infused debate), ToC (Weinzierl and Harabagiu, 2024) (counterfactual prompt tree), and MPVStance (Zhang et al., 2025b) (multi-perspective verification); (4) **Fine-tuning LLMs:** FACTUAL (Li et al., 2025a) (counterfactual data augmentation) and MRGKD (Wang et al., 2025a) (multi-agent reasoning graph distillation). Our experiments employ Qwen2.5-7B-Instruct. More experimental setup details are provided in Appendix § B.2.

4.2 Main Results

4.2.1 In-Target Stance Detection

We first present the results on two public datasets under the in-target setup. The results of these experiments are illustrated in Table 1 and Table 2. **Methods based on fine-tuning PLMs**, typified by GLAN, excel in feature extraction but suffer from inadequate generalization. Their inability to effectively decipher implicit stances and complex logical relationships embedded in conversational contexts constrains their overall per-

Method	ID	GT	SM	SS	Overall
BERT	39.59	40.74	41.54	47.40	42.32
TTS	35.39	33.87	41.20	46.11	39.14
GLAN	41.28	38.93	47.34	46.65	43.55
GPT-3.5-turbo	37.83	36.64	40.33	47.43	40.56
GPT4	41.29	39.22	47.76	49.18	44.37
DeepSeek	33.44	36.35	41.86	48.47	40.03
Doubao-pro	33.10	39.17	47.79	47.72	41.95
COLA	39.97	39.43	46.19	45.73	42.83
ToC	41.06	41.25	47.72	48.12	44.54
MPVStance	42.52	<u>41.75</u>	47.83	49.55	45.41
FACTUAL	40.34	39.94	43.01	42.26	41.39
MRGKD	<u>43.87</u>	40.83	<u>48.35</u>	<u>51.56</u>	<u>46.15</u>
ExpCSD (ours)	44.24	42.91	50.04	53.67	47.72

Table 2: Results (%) on SRQ dataset in in-target setting. Best scores are in **bold**. The suboptimal scores are underlined. The significance tests of our ExpCSD over MRGKD and MPVStance (with $p < 0.05$).

formance. For **LLMs utilizing in-context learning**, such as GPT-4, while they demonstrate certain advantages on specific datasets, inherent biases from their pre-training phases persist, resulting in subpar performance across most datasets. **Prompt engineering approaches** like COLD and ToC alleviate model biases through multi-agent debates and multi-stance rationales comparisons. Though these strategies mitigate partial biases, applying multi-step deliberation to all samples induced overthinking on neutral instances, resulting in performance on some topics falling below that of vanilla LLMs. **Approaches involving fine-tuning LLMs**, such as MRGKD, strive to rectify biases by adjusting internal parameters. However, Limited training data leaves residual biases in these models and leading to performance inferior to our ExpCSD. In contrast, ExpCSD robustly identifies biases and applies targeted debiasing, achieving SOTA on two public datasets.

4.2.2 Cross-Target Stance Detection

We conducted cross-target stance detection experiments on the MT-CSD dataset. The models were first trained and validated on a source target and then tested on a destination target, covering all available targets in the dataset. Since the dataset spans three distinct domains, namely cryptocurrency (Bitcoin), business (SpaceX and Tesla), and politics (Biden and Trump), we designed cross-target experiments both within the same domain and across different domains. As shown in Table 5, ExpCSD performs best in in-domain experiments and is sometimes inferior to MRGKD in cross-

Methods	Bitcoin	Tesla	SpaceX	Biden	Trump	Overall
ExpCSD	59.77	60.37	61.33	44.80	51.29	55.51
w/o Exp	48.35	41.07	51.18	28.74	42.36	42.34
w/o DiscrimDebias	49.28	43.75	51.79	30.34	38.99	42.83
w/o PathR	54.31	56.26	55.63	42.34	46.93	51.09
w/o BackgroundR	57.11	53.75	57.40	40.65	43.41	50.46
w/o SVS	57.76	56.32	57.86	41.27	46.83	52.01
w/o ExpFilter	56.75	58.71	60.37	37.04	41.12	50.80

Table 3: Ablation study results of ExpCSD.

domain cases. This reveals that in-domain experience achieves better bias correction than cross-domain experience.

4.3 Ablation Study

To verify the effectiveness of each component, we conduct a comprehensive ablation study (Table 3). Removing rectified experience (*w/o Exp*) significantly degrades performance, confirming the importance of experience. Removing *w/o DiscrimDebias* leads to indiscriminate experience application, with performance on some datasets even worse than *w/o Exp*, which verifies the necessity of selective debiasing. Removing path retrieval (*w/o PathR*) or background retrieval (*w/o BackgroundR*) causes significant performance drops, verifying both are indispensable for experience selection. Removing stance verbalized sampling (*w/o SVS*) leads to notable decline, confirming that correct stance reasoning is critical for quality experience generation. Finally, removing experience filtering (*w/o ExpFilter*) causes obvious degradation, as blindly adding all experiences introduces invalid experiences and harms performance.

4.4 Analysis

4.4.1 Bias and Latent thinking paths

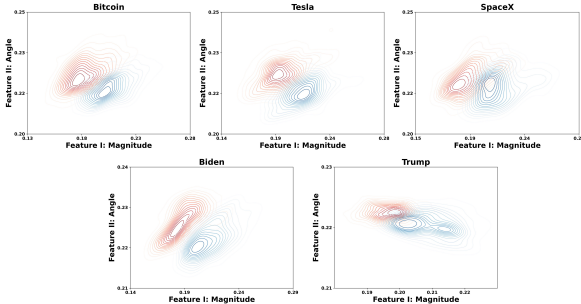


Figure 3: The geometric distribution of reasoning paths for correct and incorrect samples across five topics. Blue and Red distributions represent the correct samples and incorrect samples, respectively.

To investigate the relationship between spurious

correlation-induced bias and latent thinking paths, we conduct experiments around three questions.

Q1: Are the geometric features of thinking paths different between correct and error samples in our task? **Q2:** Is this geometric difference related to spurious correlations? **Q3:** Can geometric features distinguish correct from error paths?

To answer Q1, we split samples from five MT-CSD sub-datasets into correct and erroneous groups according to the prediction results of Qwen 2.5. We separately compute the magnitude and angle of latent reasoning paths for both groups and visualize their two-dimensional distributions in Fig 3 (see Appendix § B.4 for detailed procedures). Experimental results reveal that compared with biased reasoning characterized by shortcut and erratic tendencies, correct reasoning obtains larger magnitude implying more sufficient exploration and smaller angle reflecting better directional consistency. In the magnitude-angle space, correct and erroneous samples present stable and distinguishable distribution gaps. This validates that our proposed debiasing utility score (Eq. 6) can guide experience selection and achieve effective reasoning path rectification.

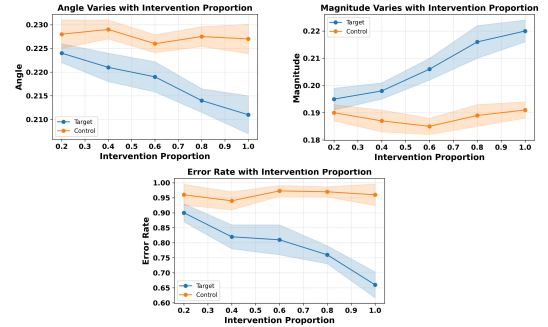


Figure 4: Curves of Error Rate, Mean Mag and Mean Ang with the Increase of Intervention Proportion.

To answer Q2, we design two intervention schemes for erroneous samples (see Appendix §B.5 for details): **Target:** Semantically invariant replacement of spurious-correlation-related words to eliminate misleading cues based on the counterfactual construction paradigm of FACTUAL (Li et al., 2025b). **Control:** Semantics-consistent random replacement of irrelevant words. As shown in Figure 4, with the increase of intervention sample proportion, the target group obtains a lower error rate, larger path magnitude and smaller excessive deflection angle, which eliminates spurious correlations and enhances latent space exploration. Insignificant changes in the control group. Slight

control group changes verify geometric variations come from spurious correlation suppression rather than lexical replacement.

To answer Q3 on judging model reasoning errors via geometric features, we report the confusion matrix of our classification method based on the latent thinking path geometric feature library in Table 6 and Table 7. More details are provided in Appendix § B.6. It obtains F1 scores of 0.89 on MT-CSD and 0.82 on SRQ, verifying that such geometric features can effectively distinguish correct and erroneous reasoning.

4.4.2 Impact of Stance Verbalized Sampling

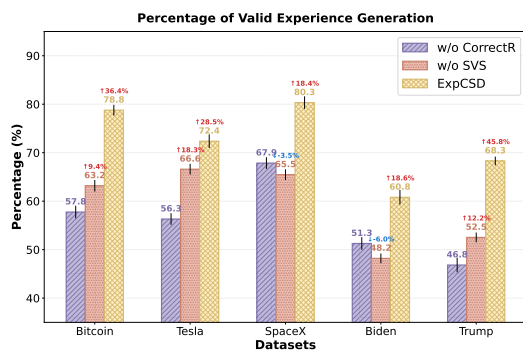


Figure 5: Effective Experience Proportion under the Difference Strategy. The numbers above the bars represent the specific proportions of effective experiences generated. ↑ and ↓ indicate percentage increase and decrease relative to the *w/o CorrectR* setting, respectively.

To explore the impact of stance verbalized sampling on valid experience generation, we conduct ablation experiments. Here, valid experiences denote experiences that help models correct biased stances. We conducted controlled ablation experiments with two variants: *w/o CorrectR* (Only high-temperature sampling without correct rationales) and *w/o SVS* (reverse rationale generation from labels (Zou et al., 2025)). As shown in Fig 5, ExpCSD increases valid experiences by 29% on average. Our stance verbalized sampling outperforms direct label-to-rationale inference by avoiding *sycophancy* (Naddaf, 2025) and selecting rationales from the model’s probability distribution. Detailed analysis is in Appendix B.7.

4.5 Case Study and Time Complexity

Case Study: Through case analysis, we show how ExpCSD uses similar bias correction experiences to debias conversational stance detection. As shown in Figure 12, our method first conducts direct reasoning, and retrieves relevant experiences

for debiasing once biases are detected. The retrieved experiences match context-induced biases well, verifying the validity of similarity-based experience selection, which also achieves strong generalization for LLM bias correction.

Time Complexity: To verify the time efficiency of our method, as shown in Table 8, we compare its inference speed with prompt-based methods including MPVStance(Zhang et al., 2025b), ToC(Weinzierl and Harabagiu, 2024), and COLA(Lan et al., 2024). Unlike these methods, which apply elaborate chain-of-thought reasoning to all cases, ours performs debiasing only when needed, thereby improving average inference speed. See Appendix § B.9 for more details.

4.6 Generalization Analysis: Framework Robustness Across LLM Backbones

To assess framework generalizability, we evaluate ExpCSD on five diverse open-source LLMs: Qwen2.5-7B-Instruct (Yang et al., 2024b), LLaMA3-8B (AI@Meta, 2024), Mistral-7B (Jiang et al., 2023), Gemma-3-4B-IT (Kamath et al., 2025), and Qwen-2-7B-Instruct (Yang et al., 2024a). As shown in Fig. 13, results demonstrate consistent performance across different architectures and scales, with Qwen2.5-7B-Instruct achieving the best overall F1 (55.51%). This confirms that our framework generalizes robustly across LLM backbones with varying parameter scales and design paradigms.

5 Conclusion

In this paper, we present ExpCSD, an experience-based debiasing framework for conversational stance detection, which models latent thinking paths from LLM hidden-state dynamics to detect biased reasoning. To generate high-quality debiasing experiences, we introduce stance verbalized sampling, which elicits diverse and correct reasoning rationales, and apply a two-stage filtering mechanism based on debiasing utility scores to remove ineffective experiences. Finally, a thinking path similaritybased selective debiasing strategy injects only the most relevant experiences, avoiding unnecessary reasoning for neutral cases while mitigating spurious correlation biases without LLM fine-tuning. Experiments on two public stance detection benchmarks show that ExpCSD outperforms baselines, validating our experience-driven debiasing framework.

Limitations

Despite the strong performance of ExpCSD, several limitations present opportunities for future work.

- **Access to Latent Representations.** ExpCSD detects biased reasoning by modeling latent thinking paths from hidden-state representations. This design enables more fine-grained bias identification than surface-level prompts or rationales, but it assumes access to model internals. Therefore, direct application to fully black-box LLM APIs may require additional approximation strategies, such as using output confidence, rationale consistency, or response variation as surrogate indicators. Exploring black-box-compatible thinking-path estimation is an interesting direction for future work.
- **Dependence on Backbone LLM Capabilities.** The effectiveness of ExpCSD is still influenced by the underlying backbone LLM. If the model fails to capture conversational context or produces highly unstable initial predictions, the quality of thinking-path assessment and subsequent experience-based debiasing may be affected. Nevertheless, ExpCSD is designed as a parameter-free framework and can be applied on top of different backbone LLMs, making it complementary to future advances in stronger and more reliable language models.
- **Coverage of the Experience Library.** ExpCSD relies on historically collected debiasing experiences to correct biased predictions. Although our two-stage filtering strategy removes ineffective experiences and retains those with demonstrated debiasing utility, the benefit of retrieval may decrease when test samples involve substantially different targets, domains, or conversational styles that are underrepresented in the library. Continually expanding and updating the experience library across broader scenarios may further improve generalization.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments and helpful suggestions. This work was supported by the

following funding sources: The National Key Research and Development Program of China (Grant No. 2023YFC3306103).

References

- AI@Meta. 2024. [Llama 3 model card](#).
- Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025. [Training-free group relative policy optimization](#). *Preprint*, arXiv:2510.08191.
- Renfei Dang, Zhening Li, Shujian Huang, and Jiajun Chen. 2025. [The first impression problem: Internal bias triggers overthinking in reasoning models](#). *Preprint*, arXiv:2505.16448.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). In *North American Chapter of the Association for Computational Linguistics*.
- Hanwen Du, Yuxin Dong, and Xia Ning. 2025. Latent thinking optimization: Your latent reasoning language model secretly encodes reward signals in its latent thoughts. *arXiv preprint arXiv:2509.26314*.
- Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *ArXiv*, abs/2310.06825.
- Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram  e, Morgane Rivi  re, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, and 191 others. 2025. [Gemma 3 technical report](#). *ArXiv*, abs/2503.19786.
- Xiaochong Lan, Chen Gao, Depeng Jin, and Yong Li. 2024. Stance detection with collaborative role-infused llm-based agents. In *Proceedings of the international AAAI conference on web and social media*, volume 18, pages 891–903.
- Ang Li, Jingqian Zhao, Bin Liang, Lin Gui, Hui Wang, Xi Zeng, Xingwei Liang, Kam-Fai Wong, and Ruifeng Xu. 2025a. Mitigating biases of large language models in stance detection with counterfactual augmented calibration. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7075–7092.

- Ang Li, Jingqian Zhao, Bin Liang, Lin Gui, Hui Wang, Xi Zeng, Xingwei Liang, Kam-Fai Wong, and Ruifeng Xu. 2025b. [Mitigating biases of large language models in stance detection with counterfactual augmented calibration](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7075–7092, Albuquerque, New Mexico. Association for Computational Linguistics.
- Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021. [P-stance: A large dataset for stance detection in political domain](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365, Online. Association for Computational Linguistics.
- Yingjie Li, Chenye Zhao, and Cornelia Caragea. 2023a. [Tts: A target-based teacher-student framework for zero-shot stance detection](#). In *Proceedings of the ACM web conference 2023*, pages 1500–1509.
- Yupeng Li, Haorui He, Shaonan Wang, Francis CM Lau, and Yunya Song. 2023b. [Improved target-specific stance detection on social media platforms by delving into conversation threads](#). *IEEE Transactions on Computational Social Systems*, 10(6):3031–3042.
- Miryam Naddaf. 2025. [Ai chatbots are sycophants and its harming science](#). *Nature*, 647:13.
- Fuqiang Niu, Zebang Cheng, Xianghua Fu, Xiaojiang Peng, Genan Dai, Yin Chen, Hu Huang, and Bowen Zhang. 2024a. [Multimodal multi-turn conversation stance detection: A challenge dataset and effective model](#). In *Proceedings of the 32nd ACM International Conference on Multimedia, MM '24*, page 38673876. Association for Computing Machinery.
- Fuqiang Niu, Min Yang, Ang Li, Baoquan Zhang, Xiaojiang Peng, and Bowen Zhang. 2024b. [A challenge dataset and effective models for conversational stance detection](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 122–132.
- Ivo Petrov, Jasper Dekoninck, and Martin Vechev. 2025. [Brokenmath: A benchmark for sycophancy in theorem proving with llms](#). *arXiv preprint arXiv:2510.04721*.
- Mark D. Rintoul and Andrew T. Wilson. 2015. [Trajectory analysis via a geometric feature space approach](#). *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8.
- Ramon Villa-Cox, Sumeet Kumar, Matthew Babcock, and Kathleen M Carley. 2020. [Stance in replies and quotes \(srq\): A new dataset for learning stance in twitter conversations](#). *arXiv preprint arXiv:2006.00691*.
- Bingbing Wang, Zhixin Bai, Qianlong Wang, Jingjie Lin, Min Yang, Xi Zeng, and Ruifeng Xu. 2025a. [More than just a conversation: A multi-agent reasoning graph knowledge distillation for conversational stance detection](#). *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- Xiaolong Wang, Yile Wang, Sijie Cheng, Peng Li, and Yang Liu. 2024a. [DEEM: Dynamic experienced expert modeling for stance detection](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4530–4541.
- Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, and Rui Wang. 2025b. [Latent space chain-of-embedding enables output-free LLM self-evaluation](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.
- Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, Zhuosheng Zhang, and Rui Wang. 2024b. [Embedding trajectory for out-of-distribution detection in mathematical reasoning](#). *Advances in Neural Information Processing Systems* 37.
- Maxwell Weinzierl and Sanda M. Harabagiu. 2024. [Tree-of-counterfactual prompting for zero-shot stance detection](#). In *Annual Meeting of the Association for Computational Linguistics*.
- 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, and 39 others. 2024a. [Qwen2 technical report](#). *ArXiv*, abs/2407.10671.
- Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024b. [Qwen2.5 technical report](#). *ArXiv*, abs/2412.15115.
- Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. 2025. [Physics of language models: Part 2.1, grade-school math and the hidden reasoning process](#). In *International Conference on Representation Learning*, volume 2025, pages 97699–97709.
- Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyang Shi. 2025a. [Verbalized sampling: How to mitigate mode collapse and unlock llm diversity](#). *Preprint*, arXiv:2510.01171.
- Zhaodan Zhang, Zhao Zhang, Jin Zhang, Hui Xu, and Xueqi Cheng. 2025b. [MPVStance: Mitigating hallucinations in stance detection with multi-perspective verification](#). In *Proceedings of the 63rd Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1053–1067.

Minghui Zou, Ronghui Guo, Sai Zhang, Xiaowang Zhang, and Zhiyong Feng. 2025. TAT: Improving stance detection on social media through thought alignment with llms. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1558–1562.

A Full Prompt Templates

This chapter comprehensively introduces all the prompt templates proposed in this paper. Specifically, the naive prompt template for conversational stance detection p_{naive} is shown in Figure 6; the complete version of the experience generation prompt template p_{exp} is presented in Figure 7; the full prompt template of Prompt of Stance Verbalized Sampling p_{svs} is illustrated in Figure 8; and the complete experience-debiasing based prompt template $p_{\text{debiasing}}$ is displayed in Figure 9. The spurious cues replacement template for generating counterfactual data in FACTUAL (Li et al., 2025b) is shown in Figure 10.

B Full Experimental Data

B.1 Dataset Statistics

To provide comprehensive context for our experimental results, we report the detailed statistics of the two datasets used in our evaluation in Table 4. The table shows the sample distribution across the training, validation, and test sets. We evaluate ExpCSD on two diverse datasets. SRQ (Villa-Cox et al., 2020) is a Twitter stance detection dataset consisting of quotes and replies. It covers four targets, including Iran Deal (ID), General Terms (GT), Student Marches (SM), and Santa Fe Shooting (SS), and contains six stance categories: Explicit Support, Implicit Support, Comment, Implicit Denial, Explicit Denial, and Queries. MT-CSD (Niu et al., 2024b) is a large-scale conversational stance dataset spanning five target topics: Bitcoin, Tesla, SpaceX, Biden, and Trump. It includes three stance categories: favor, against, and none. Notably, 75.99% of its samples contain more than four conversation turns, which increases the complexity and realism of the evaluation. Following MRGKD (Wang et al., 2025a)

and GLAN (Niu et al., 2024b), we use their released dataset splits for SRQ and MT-CSD. The two publicly available datasets used in this work are released under academic licenses (e.g., non-commercial research-only terms). Our usage complies with their intended license agreements and terms of service.

Dataset	Topic	# Train	# Valid	# Test
MT-CSD	Bitcoin	2283	431	538
	Tesla	2552	536	540
	SpaceX	1440	268	321
	Biden	2017	417	411
	Trump	2633	557	561
SRQ	ID	872	187	187
	GT	1414	303	303
	SM	516	110	111
	SS	596	128	128

Table 4: Detailed statistics of the experimental data, showing the number of samples in the training, validation, and test sets for each target.

B.2 Implementation details:

The specific model employed in our experiments is Qwen2.5-7B-Instruct. During the experience generation phase, we generate $n = 5$ reasoning traces and experiences for each biased sample. For the selection of debiasing experiences, we set $K = 3$ in the top- K selection strategy. We adopt the same dataset split as that used in MRGKD (Niu et al., 2024b) and GLAN (Niu et al., 2024b). The similarity weight α is set to 0.75. All experiments are conducted on a single NVIDIA A800 GPU with 80GB of memory. Refer to Appendix § B.9 for inference latency. Each group of data is run 5 times, and the average value is adopted.

B.3 Cross-target stance detection

We conducted cross-target stance detection experiments on the MT-CSD dataset. The models were first trained and validated on a source target and then tested on a destination target, covering all available targets in the dataset. Since the dataset spans three distinct domains, namely cryptocurrency (Bitcoin), business (SpaceX and Tesla), and politics (Biden and Trump), we designed cross-target experiments both within the same domain and across different domains. As shown in Table 5, in same-domain experiments, the ExpCSD model outperforms other models, whereas

Prompt of Conversation Stance Detection

Instruction: Analyze the stance of the last user's utterance in the following multi-turn conversation list towards **{target}**.

Stance Definitions:

favor: A stance of support, affirmation, or approval towards **{target}**, including positive evaluation, etc.

against: A stance of opposition, negation, or criticism towards **{target}**, including negative evaluation, etc.

none: A neutral stance with no support or opposition, unclear attitude, or being irrelevant to **{target}**.

Requirements:

1. First, analyze the relationship between the last user's utterance and **{target}**, and judge the basis for the stance by combining it with the historical conversation.

2. The output format must satisfy the following requirements:

- The reasoning process should be within the `<think></think>` tag.

- The final stance result should be within the `<answer></answer>` tag, and must be one of favor, against, none.

Conversation List:

{conversation_i}

Output example:

target: SpaceX

Conversation: [user 1: SpaceX Starship explodes shortly after launch'] ==> `<think>`The last utterance only states the objective fact that SpaceX's Starship exploded shortly after launch, with no explicit subjective evaluations, support, or opposition. Thus, its stance is "none".`</think><answer>`none`</answer>`

Figure 6: Prompt of Conversation Stance Detection.

Prompt of Experience Generation

Instruction: You are a professional expert in reasoning trajectory analysis. **Your task is to extract {n} generalizable and error-correctable experiences from the flaws in multiple reasoning trajectories for the following stance detection task.**, and strictly comply with the subsequent steps and output format.

Step 1: Trajectory Analysis (MUST complete all 4 sub-items)

1. Successful steps: Identify key correct decisions and insights from the provided trajectories.

2. Errors: Pinpoint exactly where and why the reasoning went wrong (compare trajectories with groundtruth).

3. Strategies: Note important thinking patterns or strategies that were used OR missed in the trajectories.

4. Failure Root Cause: Analyze why flawed reasoning trajectories fail.

Step 2: Experience Requirements (MUST satisfy all 3 rules)

1. Start with the applicable background of the experience (related to the problem type, e.g., stance detection).

2. Focus on strategic thinking patterns, NOT specific calculations or sentence details.

3. Emphasize decision points that can be applied to similar problems.

Step 3: Output Format (MUST strictly comply, VIOLATION WILL BE REJECTED)

- The reasoning process (≤ 3 sentences) must be wrapped in `<think>` tags.

- The final experience must be wrapped in `<experience>` tags.

- NO other sections, headings, or explanations outside the tags.

Problem: {conversation stance problem *q*}

trajectories: {[trajectory₁, ..., trajectory_n]}

Groundtruth: {label}

Correct Output Example

`<think>`Analyze the trajectory's error in ignoring implicit sentiment, compare with groundtruth, and summarize the core adjustment direction.`</think>` `<experience>`[background]When judging stance in conversations, beyond explicit expressions: [strategy]we need to infer implicit sentiment from context-related comments.`</experience>`

Figure 7: Prompt of Experience Generation.

Prompt of Stance Verbalized Sampling

Instruction: You are a helpful assistant. For the following multi-party conversation stance detection task, you must generate 3 possible responses that strictly correspond to the three stances: favor, against, and none respectively. Each response should be placed individually within a <response> tag. Every response must include a <answer>(final stance) and a numerical <probability> (probability value). **Please sample at random from the tails of the distribution, such that the probability of each response is less than 0.10.**

Stance Detection Task:

Judge the last user's statement in the following multi-party conversation regarding {target}'s position, and output one of favor, against, or none.

Conversation List: {conversation_i}

Correct Output Example:

target: Tesla

Conversation: ["user 1: \$140k Tesla quality",

"user 2 : My friend just bought a model X, paid over \$100,000 and the brake lights dont even match when they light up. Left side its the bottom strip. Right side its a bulb in the middle.",

"user 3: Lol that's hilarious"] ==>

<response>The last user's comment "Lol that's hilarious" is a sarcastic response to the high-price yet low-quality issue of Tesla Model X described by User 2. The tone mocks Teslas poor quality control, expressing disapproval of the brands product reliability.</response><answer>against</answer><probability>0.05</probability>

Figure 8: Prompt of Stance Verbalized Sampling.

Prompt of Experience-based Debiasing

Instruction: Based on the following stance definitions and stance detection experiences, analyze the stance of the last user's utterance in the multi-turn conversation list towards {target}, and output the reasoning process and result.

Experiences: {E_{exp} from Experience Dictionary Retrieval.}

Stance Definitions: {favor: ..., against: ..., none: ...}

Requirements:

1. First, analyze the relationship between the last user's utterance and {target}, and judge the basis for the stance by combining it with the historical conversation and **stance detection experiences**.
2. The output format must satisfy the following requirements:
 - The reasoning process should be within the <think></think> tag.
 - The final stance result should be within the <answer></answer> tag, and must be one of favor, against, none.

Conversation List:

{conversation_i}

Output example:

Experiences:[Background]The user merely provides objective factual reports with no subjective praise or criticism of the target.[Strategy] Classify such utterances as none (neutral) instead of favor or against.

Conversation:['user 1: SpaceX Starship explodes shortly after launch'] ==>

<think>Referring to the given experience, this utterance only gives an objective factual report about the SpaceX Starship incident, without any subjective praise or criticism of the target. According to the corresponding strategy, it should be classified as none rather than favor or against.</think><answer>none</answer>

Figure 9: Prompt of Experience-based Debiasing.

in cross-domain experiments, its performance is occasionally slightly inferior to that of MRGKD. This indicates that the bias correction effect of experience within the same domain is superior to that derived from different domains.

B.4 Thinking Path Feature Distribution

Using vanilla Qwen2.5-7B, we predicted on the full datasets of the five topics, splitting correct/incorrect samples into feature sets $V_+ = \{(\text{Mag}_i^+, \text{Ang}_i^+)\}_{i=1}^{n_+}$ and $V_- = \{(\text{Mag}_i^-, \text{Ang}_i^-)\}_{i=1}^{n_-}$. Here, Mag_i and Ang_i

Prompt for Spurious Cue Replacement

[sentence]: {sentence}
 [target]: {target}
 [stance]: {stance}
 [spurious cues]: {spurious_cues}

The [sentence] expresses a {stance} stance toward the [target]. The following words or phrases in the [sentence] have been identified as spurious surface cues that may mislead a language model into making incorrect predictions: [spurious cues].

Please rewrite the [sentence] according to the following requirements:

1. Replace, paraphrase, or neutralize ALL identified [spurious cues] so that they no longer provide the same misleading surface signal.
2. If a [spurious cue] is sentiment-bearing or evaluative, replace it with a neutral, factual, or attribution-based expression rather than another sentiment-bearing or evaluative word.
3. Preserve the original factual information and event structure as much as possible, including who did what, to whom, when, and where.
4. Preserve the original {stance} stance toward the [target] without making it more positive, negative, or ambiguous.
5. Do **NOT** introduce any new sentiment-bearing words, evaluative language, stance indicators, or new entity associations that were not present in the original sentence.
6. If removing a [spurious cue] would delete important factual content, generalize it into a neutral expression instead of omitting it entirely.
7. The rewritten sentence should read naturally and fluently.

Output the rewritten sentence only, without any explanation.

Figure 10: Prompt template for spurious cue replacement. Blue fields ({sentence}, {target}, {stance}, {spurious_cues}) are filled with the corresponding values from each sample.

Target	BERT	TTS	GLAN	MRGKD	ExpCSD
within the same domain					
SpaceX→Tesla	30.06	38.78	40.08	46.31	50.22
Tesla→SpaceX	37.32	40.06	40.85	52.49	56.04
Biden→Trump	24.87	30.41	31.56	34.85	46.35
Trump→Biden	20.34	28.87	30.10	36.46	41.62
across dissimilar domains					
Bitcoin→SpaceX	40.26	39.38	40.56	54.16	55.74
Bitcoin→Tesla	34.84	36.37	38.49	46.07	48.92
Bitcoin→Biden	28.34	32.70	28.78	40.09	40.31
Bitcoin→Trump	32.45	32.97	30.12	35.57	40.48
Tesla→Bitcoin	27.65	37.35	30.18	51.17	54.60
Trump→Bitcoin	33.21	35.29	29.39	53.04	51.18
SpaceX→Trump	36.08	39.58	32.40	43.81	44.92
Trump→SpaceX	22.89	26.27	27.73	53.10	52.41

Table 5: Comparison of different models for cross-target stance detection. Best scores are in **bold**. The significance tests of our ExpCSD over MRGKD (with $p < 0.05$).

denote the magnitude and angle of the thinking path for sample i (Equation 2). We estimate each sets thinking-path feature distribution via 2D Gaussian kernel density estimation (Equation 8), as shown in Fig. 11. Unbiased thinking path involves more extensive exploration (larger magnitude) while maintaining higher directional consistency (smaller angle) on semantic compared to the

shortcut-like, erratic nature of biased reasoning. Accordingly, the debiasing utility score (Eq.6) can guide experience selection for effective path correction, supporting the validity of distinguishing biased reasoning based on path similarity.

$$f_V(\text{Mag}, \text{Ang}) = \frac{1}{nh^2} \sum_{i=1}^n \frac{1}{2\pi} \exp \left\{ -\frac{1}{2h^2} [(\text{Mag} - \text{Mag}_i)^2 + (\text{Ang} - \text{Ang}_i)^2] \right\} \quad (8)$$

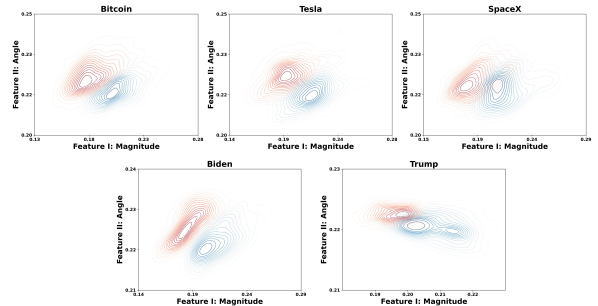


Figure 11: The thinking path feature distribution of correct and incorrect sample sets in three topics. Blue and Red distributions represent the correct samples and incorrect samples, respectively.

B.5 Spurious Feature Intervention

To investigate the source of the geometric differences between the two types of latent thinking

paths, we designed a controlled intervention experiment. Specifically, we constructed two intervention conditions based on samples that were incorrectly predicted by the model.

- **Target group.** We first manually selected all 967 erroneous samples from the training data under the Tesla topic. Five annotators were then invited to identify spurious surface cues that could mislead the large language model into incorrect thinking, including sentiment-bearing words and entity mentions. In order to identify spurious surface cues that could mislead large language models into incorrect reasoning, we manually annotated the data with the help of five volunteers. Specifically, we selected five volunteers between the ages of 20 and 30 with backgrounds in stance detection. Among them, there are three males and two females. Three are PhD students, and two are masters students. We created annotation guidelines, requiring each volunteer to identify misleading surface cues in the input texts, including sentiment-bearing words and entity mentions. The human annotation results were then used as the reference for further analysis of shortcut reasoning in large language models. The annotation instructions are shown in Table 9. Based on these annotations, we followed the prompting strategy used by the FACTUAL (Li et al., 2025b) for generating counterfactual data, and replaced the identified spurious cues while preserving the original semantics as much as possible, as shown in Figure 10. This process yielded the target intervention sentences.
- **Control group.** To rule out the possibility that the observed effects were caused merely by lexical perturbations, we constructed a control condition using random word replacement under the same semantic-preservation constraint. The control sentences were also generated using the same counterfactual prompting strategy as in the target group.

We then progressively increased the intervention intensity according to the proportion of intervention sentences among all erroneous samples, and evaluated the model error rate as well as the corresponding geometric features under different intervention ratios. Each group of experiments was

repeated five times; the curves report the mean values over the 5 runs, and the shaded regions represent standard variances of all samples. The results are shown in Fig 4. As the proportion of intervened samples increased in the target group, the model error rate consistently decreased. Meanwhile, the average magnitude of the latent reasoning paths increased, whereas the angle values decreased. This suggests that, as spurious surface cues were gradually weakened or removed, the model became less affected by bias-inducing distractions, and the angular deviation caused by spurious correlations was also reduced. At the same time, the models trajectory in the latent space exhibited greater exploratory behavior, reflected by the increased magnitude, which ultimately led to a lower error rate.

In contrast, the same trends were much weaker in the control group. This indicates that random lexical replacement alone cannot induce behavioral and geometric changes comparable to those produced by the targeted intervention. Overall, these results suggest that the observed geometric differences are highly consistent with the mechanism of suppressing spurious associations, thereby providing experimental evidence for the origin of the geometric differences between the two types of latent reasoning paths.

B.6 Classification Based on Biased and Unbiased Thinking Path Libraries

To verify whether the geometric features of latent paths can effectively distinguish incorrect reasoning from correct reasoning, this paper adopts a classification method based on comparison with thinking path libraries. Discriminative experiments are conducted on the test sets of the MT-CSD and SRQ datasets, and the confusion matrices of the model reasoning correctness discrimination results are reported. During the experiments, test data from multiple sub-datasets under each dataset are merged. The MT-CSD dataset contains a total of 2,371 test samples, while the SRQ dataset contains a total of 729 test samples.

This paper standardizes the definitions of positive and negative samples in the discriminative experiments: *cases judged by the discrimination method as correct model reasoning are defined as the predicted positive class, and cases where the model’s reasoning is actually correct are defined as the true positive class*. In both datasets, samples with actually correct reasoning account

Predicted \ Actual	Actually Incorrect	Actually Correct
Predicted Incorrect	632 (TN)	173 (FN)
Predicted Correct	158 (FP)	1408 (TP)

Table 6: Confusion matrix of the MT-CSD dataset. Total samples: 2,371; actually correct: 1,581; actually incorrect: 790; $F1 = 0.8948$.

Predicted \ Actual	Actually Incorrect	Actually Correct
Predicted Incorrect	177 (TN)	92 (FN)
Predicted Correct	76 (FP)	384 (TP)

Table 7: Confusion matrix of the SRQ dataset. Total samples: 729; actually correct: 476; actually incorrect: 253; $F1 = 0.8205$.

for approximately two-thirds, while samples with actually incorrect reasoning account for approximately one-third. Under this definition, a false positive refers to a case where the model’s reasoning is actually incorrect but is mistakenly judged by the discrimination method as correct; a false negative refers to a case where the model’s reasoning is actually correct but is mistakenly judged by the discrimination method as incorrect.

The confusion matrix statistics for the two datasets are shown in Table 6 and Table 7. In the MT-CSD dataset, there are 1,408 true positive samples, 158 false positive samples, 632 true negative samples, and 173 false negative samples. The F1 value for this dataset is approximately 0.8948. In the SRQ dataset, there are 394 true positive samples, 66 false positive samples, 177 true negative samples, and 92 false negative samples. Similarly, the F1 value is approximately 0.8330. For ease of presentation and comparative analysis in the paper, the F1 value of the MT-CSD dataset is rounded to two decimal places as 0.89, and the F1 value of the SRQ dataset is rounded to two decimal places as 0.83.

The confusion matrix results show that the proposed method produces relatively few misclassified samples on both datasets and demonstrates good recognition capability for both correct and incorrect model reasoning. The method exhibits good discriminative stability and no obvious class bias. The experimental results indicate that the geometric features of thought paths can effectively capture implicit differences in the reasoning process, accurately distinguish whether model reasoning is correct or incorrect, and fully validate the effectiveness and feasibility of using these geometric features for reasoning quality discrimination.

B.7 Impact of Stance Verbalized Sampling on Experience Generation

We designed controlled ablation experiments to study how correct rationales and stance verbalized sampling affect the generation of valid experiences, where valid experiences are those that help the model correct a biased sample’s stance:

- “w/o *CorrectR*” variant extracts experiences solely from multiple high-temperature sampled outputs. Under the impact of typicality bias, the model relies only on its own generated biased stance rationales, which are combined with the ground-truth labels to guide valid experience generation.
- “w/o *SVS*”: Instead of obtaining correct rationales through stance verbalized sampling, this variant directly provides the ground truth to the model and prompts it to generate a corresponding rationale in reverse, following an approach similar to TAT (Zou et al., 2025).

Since each experience generation process yields multiple candidate experiences, we directly calculate the proportion of those that are valid. The specific formula is as follows:

$$P_{valid} = \frac{1}{N} \sum_j^N \frac{|\mathcal{E}_{valid}^j|}{n} \quad (9)$$

where N is the number of samples in the biased Library, n is the number of experiences generated per sample ($n = 5$ in our experiments), and $|\mathcal{E}_{valid}^j|$ counts valid experiences for the j -th sample. The results are shown in Figure 5. Compared to w/o *CorrectR*, ExpCSD with correct rationales increases the proportion of valid experiences by an average of 29% across datasets. Additionally, the strategy of generating correct rationales directly from labels rather than through SVS increases the proportion of valid experiences on some datasets but actually produces negative effects on other valid datasets. It underperforms compared to the correct rationales obtained via stance verbalized sampling in our ExpCSD. This is likely because inferring rationales by reversing from ground-truth stance labels amplifies *sycophancy* (Naddaf, 2025; Petrov et al., 2025) in LLMs, which results in hallucinated rationales and degrades overall performance. In contrast, our stance verbalized sampling method selects based on the model’s own probability distribution, thereby generating more

Models	Core Idea	Avg. Latency (s)
COLD (Lan et al., 2024)	Multi-expert debate	~25s
ToC (Weinzierl and Harabagiu, 2024)	Counterfactual rationale generation comparison	~14s
MPVStance (Zhang et al., 2025b)	Multi-perspective verification	~32s
ExpCSD (ours)	On-demand debiasing	~5s

Table 8: Comparison of inference latency between ExpCSD and existing prompt-based debiasing methods.

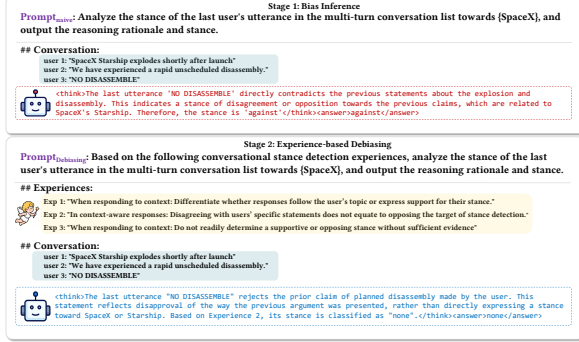


Figure 12: An case of Correcting Bias Caused by Context-Induced Stance Target Errors.

correct rationales and eliciting a greater number of valid experiences.

B.8 Case study

Through case analysis, we show how ExpCSD uses similar bias correction experiences to debias conversational stance detection. As illustrated in Figure 12, the first stage performs direct reasoning, and in the second stage, detected biases trigger the retrieval of relevant experiences for debiasing. We can find that the retrieved experiences are inherently relevant to context-induced biases, demonstrating the rationality of selecting experiences based on thinking similarity. Moreover, these experiences generalize well and effectively correct biases in LLMs.

B.9 Time Complexity

To assess the operational efficiency of the proposed on-demand debiasing framework, we conduct a comprehensive comparison on the MT-CSD dataset against state-of-the-art prompt-engineering baselines, including MPVStance (Zhang et al., 2025b), COLD (Lan et al., 2024), and ToC (Weinzierl and Harabagiu, 2024). as shown in Table 8, these methods typically apply sophisticated chain-of-thought reasoning to every input instance, regardless of whether debiasing is necessary. In contrast, our framework selectively invokes empirical debiasing only when model

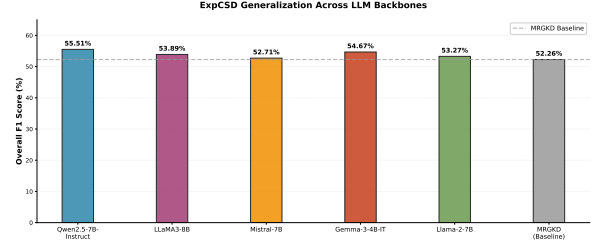


Figure 13: ExpCSD generalization performance across five LLM backbones on MT-CSD.

bias is detected at inference time. By avoiding redundant reasoning for unbiased instances, the proposed selective execution strategy substantially reduces computational overhead and leads to significantly lower average inference latency.

B.10 Generalization

To assess framework generalizability, we evaluate ExpCSD on five diverse open-source LLMs: Qwen2.5-7B-Instruct (Yang et al., 2024b), LLaMA3-8B (AI@Meta, 2024), Mistral-7B (Jiang et al., 2023), Gemma-3-4B-IT (Kamath et al., 2025), and Qwen-2-7B-Instruct (Yang et al., 2024a). As shown in Fig. 13, results demonstrate consistent performance across different architectures and scales, with Qwen2.5-7B-Instruct achieving the best overall F1 (55.51%). This confirms that our framework generalizes robustly across LLM backbones with varying parameter scales and design paradigms.

Table 9: Cue Type Reference Table with Examples

Cue Type	Definition	Annotation Criterion	Example	Spurious Cues
Sentiment-bearing Word	A word or phrase expressing strong positive or negative sentiment that the model treats as the main predictive evidence, regardless of whether it reflects the author’s actual stance.	Annotate when the model explanation explicitly uses a sentiment word as the decisive basis for prediction, while the correct label does not depend on sentiment polarity.	<i>Text:</i> “Some reviewers described Teslas vehicle as disappointing during the road test.” <i>Model rationale:</i> “The word disappointing indicates negative sentiment toward Tesla.” <i>Correct label:</i> Neutral	“disappointing”
Entity Mention	A person, organization, location, or brand name that the model associates with a specific label through prior knowledge or learned correlations, rather than reasoning from the actual content.	Annotate when the model explanation reveals that it relied on the entity name itself rather than the contextual content surrounding it.	<i>Text:</i> “Mr. Musk announced Tesla’s latest quarterly results at the shareholders’ meeting.” <i>Model rationale:</i> “Using honorific “Mr.” for Musk implies supportive attitude toward Tesla.” <i>Correct label:</i> Neutral	“Mr. Musk”
Spurious Key-word	A word or phrase that frequently co-occurs with a specific label in the training data, which the model uses as a shortcut without considering negation, contrast, or the broader context.	Annotate when the model highlights an isolated keyword as evidence while ignoring the surrounding semantics, such as negation or irony.	<i>Text:</i> “Tesla’s sales meet the record highs analysts had expected.” <i>Model rationale:</i> “The phrase ‘record highs’ is associated with positive performance, suggesting a supportive stance towards Tesla.” <i>Correct label:</i> Neutral	“record highs”