'12-Angry LLMs' - Divergence from Deliberation as Signal for Complex Stance Detection

Anonymous authors
Paper under double-blind review

Abstract

This study introduces "12-Angry LLMs," a novel annotation and classification model that leverages annotator disagreement to improve complex stance detection. Departing from traditional methods that average out divergence, we deploy a diverse panel of 12 LLMs that engage in a two-stage process: independent voting (Round A) followed by collective deliberation (Round B) when disagreement occurs. We demonstrate that the rationales generated during deliberation serve as critical signals for fine-tuning the Judge model. On the RUStance-2023 dataset, this Judge model achieves performance (F1 $\approx$ 0.81) compared with single-model baselines and standard aggregations. The approach also proves highly transferable, achieving an F1 score of 0.94 on the out-of-domain PStance dataset using few-shot prompting with jury rationale. We contribute a new dataset containing expert labels alongside full jury deliberation traces, establishing a paradigm in which model divergence is utilized as a diagnostic tool for uncertainty and interpretability rather than noise.

1 Introduction

The task of identifying whether a text is in favor of or against a specific target issue or claim is known as stance detection. It offers a systematic way to analyze viewpoints (Tao et al., 2024). This task has broad applications; for example, in the Russia-Ukraine conflict, social media has been overwhelmed with competing narratives, and stance classification helps map how users align with or oppose key claims, thereby assisting in the detection of propaganda and misinformation (Lavrouk et al., 2024; Gorrell et al., 2019; Conforti et al., 2018). Prior work has demonstrated the viability of stance classification as an effective tool for identifying misinformation across languages and communities (Lavrouk et al., 2024). More generally, automatic stance detection plays an important role in information retrieval, content moderation, and text summarization, where understanding an author's standpoint provides valuable contextual insight (Mohammad et al., 2016; Conforti et al., 2020; Carnot et al., 2023).

Determining a stance on contentious issues is often a subjective task, and annotator divergence is a well-recognized challenge. Even expert human annotators frequently disagree on stance labels owing to ambiguous language, differing cultural backgrounds, or personal biases (Li et al., 2025). Traditionally, such disagreements are treated as noise to be resolved via majority voting or averaging. However, emerging research argues that disagreement is not merely noise but a signal of genuine differences in perspective (Fleisig et al., 2023; Uma et al., 2021). For example, studies have found that annotator disagreements in labeling hate speech or rumor veracity often stem from systematic factors, such as annotators' demographics or political leanings, rather than random error (Fleisig et al., 2023). In stance datasets, relying solely on majority votes can yield inconsistent or contradictory gold labels for similar content, underscoring the limitation of blurring out minority viewpoints (Li et al., 2025). This divergence is not restricted to humans; Large Language Models (LLMs) used as annotators or classifiers may also produce varying judgments depending on model design or prompting. A single AI "judge" can exhibit inherent biases; hence, stance analysis using LLMs demands methods that account for multiple interpretations rather than one-size-fits-all predictions (Jain, 2025).

In this study, we address annotator disagreement and propose a new approach to leverage it for better stance detection. Drawing inspiration from deliberative juries (as dramatized in the movie '12 Angry Men'), we introduce a jury of LLMs for annotation. In our model, multiple independent LLMs act as 'jurors' who each cast a stance judgment before their votes are deliberated upon and aggregated to produce a final label by a fine-tuned expert LLM that acts as a 'judge. In other words, we simulated LLMs in a real-world jury system that worked together to reach a consensus decision. This multi-LLM strategy is designed to generalize the annotation process beyond any single annotator's biases, leveraging diversity to improve reliability. To evaluate this methodology, we extended the RUStance-2023 [anonymous citation] study by applying our jury model to its gold-standard dataset of social media posts about the Russia-Ukraine conflict, annotating it in parallel with domain experts and our LLM jury. We assessed the impact of incorporating jury-based labels by benchmarking stance classification performance across human annotations, individual LLM annotations, and traditional machine learning models. In summary, our contributions are as follows.

1. **Jury of LLMs.** We introduce a new methodology in which a panel of twelve large language models collaboratively determines stance labels and corresponding rationale. Inspired by the deliberative process in the film *12 Angry Men* and the formal logic of courtroom juries, we treat each LLM as an independent evaluator and aggregate its votes through structured multiround deliberation. This approach provides a generalizable model for labeling subjective data, reducing the influence of individual model bias, and capturing disagreements as signals of underlying uncertainty.
2. **Benchmarking Across Humans, LLMs, and Classical Models.** We conducted extensive experiments comparing the stance classification performance when trained on expert gold labels. We further evaluated multiple modeling approaches, including multilayer perceptrons (MLPs), zero-shot LLMs, few-shot LLMs, and fine-tuned LLMs. This study provides the first systematic comparison of human, LLM, and jury-based annotations for stance detection.
3. **Russia–Ukraine Stance and Deliberation Dataset.** We will release a dataset of social media posts on the Russia–Ukraine conflict, annotated by a jury of 12 LLMs (Juror-RoundA Dataset). In addition to the stance labels, the dataset includes detailed records of the jury's deliberation process (Juror-RoundB Dataset), enabling researchers to study the final labels and the dynamics of LLM disagreement and consensus.

Overall, our work advances the RUStance-2023 study [anonymous citation] by providing a more comprehensive benchmark and a new annotation paradigm that treats divergence as informative by combining the precision of human expertise with the scalability of LLMs in deliberative models. We aim to foster more robust stance detection models and a deeper analysis of online conflict discourse.

This paper begins with a discussion of the related work in Section 2. Section 3 presents the RUStance-2023-Gold dataset and jury-of-LLMs deliberation corpus. Section 4 details the experimental setup, including the baseline models, LLM configurations, and aggregation strategies. The experimental findings are reported and analyzed in Section 5, highlighting the quantitative performance and qualitative insights into disagreements. Final Section 6 concludes the paper with reflections on contributions and outlines possible directions for future work.

2 Related Works

Stance detection has been an active area of research in NLP for the past decade. Early benchmarks such as 'SemEval-2016 Task 6: Detecting Stance in Tweets' introduced datasets of tweets labeled as Favor, Against, or Neither towards specific targets (Mohammad et al., 2016). Subsequent datasets expanded the scope, including P-Stance with over 21,000 political tweets labeled for stance towards U.S. politicians (Li et al., 2021) and COVID-19-Stance, capturing pandemic-related debates on social media (Glandt et al., 2021). Methods have evolved from lexical and sentiment-based classifiers to deep neural models and, more recently, to fine-tuned transformers such as BERT and RoBERTa, which have become state-of-the-art on benchmarks such as SemEval and P-Stance (Li et al., 2021). Despite these advances, generalization across unseen targets remains challenging, and stance detection relies heavily on the quality and reliability of annotations.

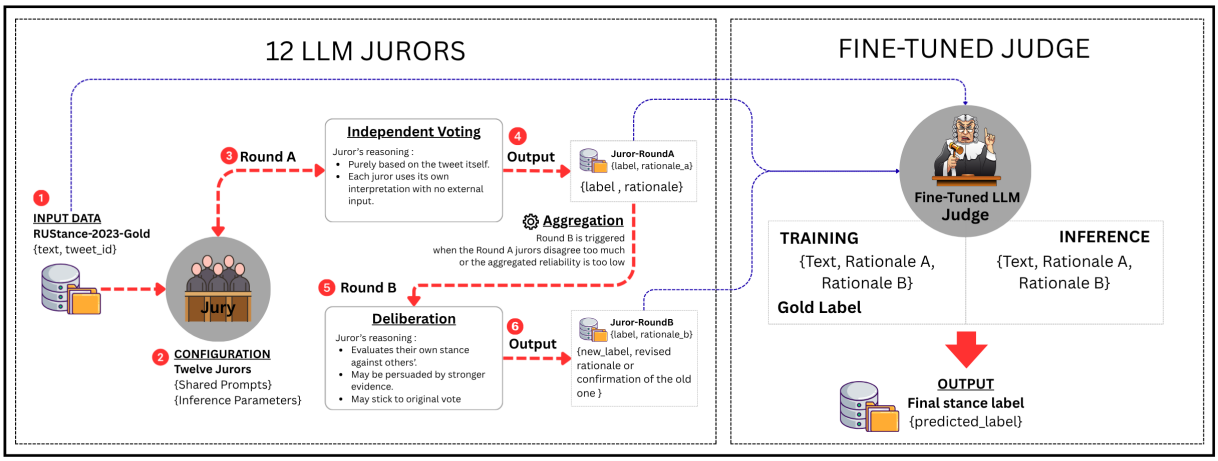


Figure 1: Process Flow of the Deliberative Jury of LLMs for Stance Detection.

A recurring difficulty in stance research is the subjectivity of the annotation. Unlike factual classification tasks, stance often depends on ambiguous cues and the annotator's interpretation, which leads to significant annotator disagreement. For example, in the SemEval-2016 stance dataset, tweets were double-annotated by a pool of eight annotators. The organizers reported an initial inter-annotator agreement of 73% (percent agreement) across all instances (Mohammad et al., 2016), reflecting the difficulty and ambiguity in many cases (e.g., sarcasm or context-dependent opinions). To ensure a reliable evaluation set, the task designers filtered out tweets with low consensus: only instances where at least five of eight annotators agreed ($\geq$ 60% agreement) were included in the official Stance Dataset. This increased the agreement on the released data to 81.9% (Mohammad et al., 2016), but it also meant that highly controversial or ambiguous examples were set aside as too difficult. Similarly, other stance annotation efforts have reported moderate levels of agreement. For example, recent studies note that Cohen's κ between annotators is often around 0.4–0.5 for stance-like subjective tasks, indicating only "moderate" agreement by conventional scales (Schiller et al., 2021). These results underscore that identifying stance can be inherently subjective; different people may legitimately interpret the exact text in favor of or against a target based on their background knowledge or beliefs. Low annotator agreement raises concerns regarding the reliability of stance labels and the validity of treating any single "ground truth" as authoritative. Conventionally, NLP datasets resolve conflicting annotations by majority vote or averaging, thereby collapsing disagreements into single labels. This practice is widespread in subjective annotation tasks (Mostafazadeh Davani et al., 2022). However, as Mostafazadeh Davani et al. (2022) argue, disagreements often "reflect annotators' individual perspectives and value" and may "encode fine distinctions that are typically overlooked" when a consensus label is imposed.

Recently, LLMs have opened new possibilities for automating annotations and LLMs are used as annotators or crowdworkers instead of humans. Gilardi et al. (2023) demonstrated that GPT-based models can outperform crowdworkers in accuracy and inter-coder reliability on multiple text annotation tasks, including stance classification. However, LLM outputs are sensitive to prompt wording and decoding randomness, raising concerns regarding consistency. To address this, self-consistent decoding (Wang et al., 2023) aggregates multiple reasoning paths from the same model to produce more reliable outputs, echoing ensemble strategies that have long been used in human annotation. Taking this ensemble approach a step further, researchers have begun experimenting with multi-agent LLM models in which several models debate, critique, or vote before making decisions. Zhao et al. (2024) shows that multi-LLM debate can significantly improve factual and logical accuracy, while Choi et al. (2025) reports that discussion phases before voting lead to stronger consensus. These approaches parallel human committees, suggesting that ensembles of models may yield more reliable outcomes than those of single annotators. In contrast to prior work, which typically attempts to minimize or eliminate disagreements, our study treats divergence as a meaningful signal. By introducing a jury of LLMs, we preserved both agreement and disagreement. Unlike the multi-agent debate model

that aims to produce a single, superior consensus, our method is designed to output the full spectrum of divergence, including structured minority reports and hung jury statistics. This enables a systematic study of when models converge and split and what these patterns reveal about subjectivity and ambiguity in stance detection.

3 Methodology

The overall methodology is illustrated in Figure 1. The methodology is divided into two parts. First jury deliberation dataset creation and second stance detection using fine-tuning the LLM model as the judge.

3.1 Jury of LLMs

We design a jury-of-LLMs process in which twelve LLMs act as independent jurors. Each juror interprets the same input text, which is a short tweet, and then produces a stance label, confidence score, and rationale for their choice. The choice of twelve balances diversity and efficiency and echoes the deliberative design popularized by *12 Angry Men* and the formal logic of courtroom juries. The jury comprised twelve large language models sourced from different providers and architectures to maximize the diversity of perspectives. The different LLMs used are listed in Table 1. The table shows that the composition ensures coverage across instruction-tuned general-purpose models (e.g., GPT-4o-mini, Claude-3.5, Gemini-2.5), specialized or hybrid models (e.g., Qwen3-Coder, WizardLM, Jamba-Mini), and large open-weight architectures (e.g., Llama-3.3-70B, Nemotron-49B, Mistral). By mixing providers and scales, the jury is deliberately designed to capture a wide range of training data biases, reasoning styles and annotation behaviors.

Table 1: Roster of LLM jurors used in the multi-round deliberation.

ID	LLM Model Variant	Parameter
J1	gpt-4o-mini-2024-07-18	undisclosed
J2	gemini-2.5-flash	undisclosed
J3	claude-3.5-sonnet	undisclosed
J4	grok-4	undisclosed
J5	llama-3.3-70b-instruct	70 B
J6	llama-3.3-nemotron-super-49b-v1	49 B
J7	deepseek-chat-v3.1	671 B, 37 B active
J8	qwen3-coder	480 B, 35 B active
J9	glm-4.5	355 B, 32 B active
J10	jamba-mini-1.7	52 B, 12 B active
J11	mistral-nemo	24 B
J12	wizardlm-2-8x22b	22 B

3.2 Multi-Round Deliberation and Aggregation

The jury of LLMs is organized into two structured rounds of annotation designed to capture both independent judgments and the dynamics of persuasion or dissent. To encourage transparent reasoning and enable analysis beyond the final label, each juror is instructed to produce a brief rationale first. The rationale captures the model's explanation for its stance decision (e.g., citing explicit support for Ukraine, detecting sarcasm, or noting ambiguous language). These rationales are later used as evidence in the deliberation phase and as sources of minority reports. Each juror is then zero-shot prompted with a rationale-then-label instruction: the model first explains its reasoning and only afterward emits a single stance label from the six-class schema. Prompts are kept identical across jurors and rounds (except for Round B, where contextual vote distributions and selected rationales are added) to avoid confounding.

1. **Round A (independent votes).** In the first round, each juror independently assigns a stance label without exposure to the views of other models. Each model receives the same input text

and a standardized system prompt that describes the stance labeling schema, which includes six possible categories: Pro-Ukraine, Pro-Russia, Pro-NATO, Anti-NATO, Neutral, and Unclear. Each juror outputs a JSON object containing: (i) **label**: predicted stance category, (ii) **confidence**: self-assessed confidence probability (0–1), (iii) **uncertain**: flag for ambiguity or sarcasm, (iv) **rationale**: brief textual justification, and (v) **evidence**: key phrases or sub-strings supporting the decision.

This produces twelve parallel judgments that reflect the unmediated diversity of perspectives across models. Thus, Round A serves as a baseline measure of individual variance, analogous to the initial polling of a human jury.

2. **Round B (deliberation with selected rationales).** The second round is executed if the results of Round A show significant uncertainty in terms of high entropy (> 1.1), small margin (< 0.2), or low reliability (< 0.9). In this phase, a 'panel summary' is automatically generated, highlighting the top two labels, representative rationales from supporting jurors, and vote counts. Each juror then revisits the same post and sees this anonymized summary and may revise or reaffirm their decisions. This is similar to structured peer deliberation in human juries, where hearing other people's arguments can help resolve borderline cases.

The system aggregates jurors' votes using one of two complementary strategies that balance simplicity, probabilistic rigor, and learned reliability. In the Majority Vote (MV) method, the ensemble selects the most frequent label as the provisional verdict, and the reliability is computed as the proportion of jurors who agree on that label. The Dawid-Skene (DS) (Dawid & Skene, 1979) method instead models each juror's accuracy probabilistically, estimating latent reliability parameters and inferring a posterior probability distribution over all possible labels. The final label is chosen as the one with the highest posterior probability, and its value provides a natural reliability score. Once the votes are combined, the system computes the entropy of the label distribution to measure disagreement, the margin between the two most frequent labels to quantify decisiveness, and the reliability of the aggregated decision. These three indicators jointly determine whether the disagreement is significant enough to trigger a second deliberation round (Round B), ensuring that only ambiguous or uncertain cases proceed to further collective reasoning.

3.3 The Judge: Fine-tuning LLM (Deliberation and Gold Label) and Inferencing

The rationales collected from Rounds A and B are concatenated and used to fine-tune an LLM model as Judge. This Judge is implemented to reproduce expert-level decisions from aggregated juror reasoning. It serves as an efficient consensus mechanism that can generate reliable stance labels for new, unseen data. This multi-agent model transforms subjective annotations into measurable and interpretable processes. It captures not only the most likely stance but also the uncertainty structure behind disagreements.

During inference, the system processes new social media posts through an automated, multistage pipeline that mirrors the jury-based training workflow. The input tweet first passes through Round A, where twelve large language models independently act as jurors, each producing a stance label and rationale without knowledge of the others' decisions. These outputs are then passed to an aggregation module using either the Majority Vote or Dawid–Skene to combine the jurors' opinions into a provisional consensus. If the system detects significant disagreement or low reliability, Round B deliberation is triggered, in which jurors reconsider their stances after reviewing an anonymized summary of collective reasoning. The aggregated rationales from both rounds, along with the post text, are then fed into the fine-tuned LLM Judge, a model trained to emulate expert consensus. The Judge synthesizes these deliberation traces and outputs the final stance label along with its reliability score. All results are stored in a Predicted Stance Dataset, which enables further stance analysis and serves as training data for future model refinements.

4 Experiments

We conducted a systematic evaluation of stance classification under three experimental configurations. The first configuration is the MLP as a judge, which includes classical neural baselines that use text embeddings and text embeddings combined with jury-provided rationales. The second Configuration, LLM (few-shot)

as a judge, and finally, the Fine-Tuned LLM as a judge. All these experiments are tested on human gold standard data with the proposed multi-LLM juror deliberation dataset.

4.1 Dataset Alignment

We used the RUStance-2023-Gold dataset (300 posts) as an expert-labeled dataset (anonymous citation). To train the meta-judging model, we aligned the multiround jury deliberation outputs with the expert-labeled ground truth to form a structured supervision dataset. The alignment process integrates the rationales from Rounds A and B for all jurors, along with the corresponding expert stance label for each post. Specifically, for each tweet_id, the script extracts every juror's explanatory rationale from both rounds, cleans the text, and concatenates them to form a composite reasoning input. This composite string represents the collective deliberation context of the post. The resulting dataset pairs (Text + Combined Rationales) are used as inputs, with the expert label as the target.

This alignment ensures that the fine-tuned Judge model learns how human experts resolve disagreements among jurors, rather than simply mimicking majority opinions. By exposing the model to both early (Round A) and reconsidered (Round B) rationales, the training data captures the full deliberation dynamics, including shifts in justification and consensus formation. Thus, the aligned dataset teaches the Judge to weigh reasoning patterns, confidence levels, and ambiguity indicators in a manner consistent with expert decisions, effectively bridging the gap between collective model reasoning and authoritative human annotation.

In addition to RUStance-2023, we applied the same alignment and experimental pipeline to three additional stance detection datasets: PStance (Li et al., 2021), SemEval-2016 Task 6 (Mohammad et al., 2016), and GWStance (Luo et al., 2020). For these datasets, we used the same jury-of-LLMs model to generate Round A and Round B rationales and aggregated them in the same manner as described above. The combined rationales were concatenated with the original text to form the input for the Judge model and baseline configurations. This ensures that all datasets are processed under a consistent experimental setup, allowing for a fair cross-domain comparison of the impact of jury deliberation across different stance detection tasks and domains.

4.2 Experiment Configuration

To evaluate the effectiveness of our proposed jury-of-LLMs model, we compared it with a range of models, including classical neural architectures, zero- or few-shot prompting with large language models, fine-tuned LLMs, and multi-model jury aggregation. These baselines provide a comprehensive view of the performance trade-offs between accuracy, stability, and interpretability.

Table 2: Comparison of experimental setup for different LLM configurations.

Aspect	Zero-Shot LLMs	Few-shot LLMs	Fine-tuned LLMs
Model	gpt-4.1-mini (OpenRouter)	gpt-4.1-mini (OpenRouter)	gpt-4.1-mini (April 2025 fine-tuned)
Input format	Text (with/without rationales)	Text + 3 curated examples per class (with/without rationales)	Text + aggregated jury rationales
Training protocol	No task-specific training	No task-specific training	Fine-tuning on RUStance-2023
Prompting strategy	Direct classification into 6 stance labels	Exemplars prepended as in-context demonstrations	Finetuning with gold stance labels
Repetition / Voting	3 repeated calls; majority vote	3 repeated calls; majority vote	Deterministic outputs (single label per instance)
Evaluation split	4-fold cross-validation	4-fold cross-validation	4-fold cross-validation

4.2.1 Configuration 1: MLP as Judge

We first established a classical baseline using a multilayer perceptron (MLP) trained on the tweet text and jury rationales. Each tweet was encoded into a 768-dimensional vector using the `all-mpnet-base-v2` sentence transformer, while the juror rationales and deliberation texts were separately embedded and averaged into an additional 768-dimensional vector. These two representations were concatenated to form a 1,536-dimensional input feature vector. The MLP consisted of two hidden layers (512 and 128 units) with ReLU

activations and dropout regularization. Training was performed using Adam optimization for 20 epochs, using a cross-entropy loss function with class weights to address label imbalance. The evaluation was conducted under a 4-fold stratified cross-validation, reporting accuracy, macro-F1, and Cohen's κ. Predictions below a confidence threshold of 0.5 were reassigned to the *Unclear* class to handle uncertainty. This setup enabled us to test whether jury rationales contribute to a machine-usable signal when integrated with textual embeddings.

4.2.2 Configuration 2: LLM (few shots) as Judge

Building on this, we evaluated large language models (LLMs) as stance classifiers without task-specific training. All experiments used gpt-4.1-mini, accessed via the OpenRouter API with deterministic decoding. In the few-shot setting, we prepended two to three curated exemplars per class from the RUStance-2023 dataset, providing additional context. Each classification was repeated three times, and the majority label was taken as the final prediction. Rationale for Rounds A and B included assessing whether exposing the models to prior explanations improved consistency.

4.2.3 Configuration 3: Fine-Tune LLM as Judge

Fine-tuned gpt-4.1-mini (April 2025) on RUStance-2023 using concatenated tweet text and aggregated jury rationales as input. Fine-tuning was evaluated under complementary protocols. We performed a four-fold stratified cross-validation to test the robustness. The model was constrained to produce one of the six stance labels. Performance is measured using accuracy, the weighted F1 score, and confusion matrices. This experiment examined whether fine-tuning with jury rationales yields more stable and reliable predictions than prompting alone does. The configurations of the LLMs are listed in Table 2.

5 Results and Analysis

This section presents the empirical results from our experimental model, comparing classical machine learning baselines, zero and few-shot LLMs, a jury of LLMs aggregation, and fine-tuned LLMs. Our analysis focuses on three dimensions: predictive performance, reliability across classes, and output interpretability.

5.1 Overall Model Performance Results

Table 3: Macro-F1 Performance Across Datasets

Model	RUStance	PStance	SemEval	GWStance	Avg Macro-F1
MLP (Text)	0.428	0.675	0.628	0.885	0.654
MLP (Text + Rationale AB)	0.475	0.819	0.697	0.929	0.730
Zero-shot LLM (Text)	0.454	0.686	0.733	0.790	0.666
Zero-shot LLM (Text + Rationale AB)	0.611	0.922	0.705	0.790	0.757
Few-shot LLM (Text + Rationale AB)	0.647	0.943	0.736	0.783	0.777
Finetuned LLM Judge (Text + Rationale AB)	**0.815**	**0.946**	**0.693**	**0.911**	**0.791**

Table 3 presents the overall Macro-F1 performance of all model configurations across four stance datasets: RUStance-2023, PStance, SemEval, and GWStance. Several important trends emerged from this comparison. First, incorporating jury deliberation rationales generally improves performance across most datasets and model configurations, particularly for RUStance-2023 and PStance, where stance classification often requires implicit reasoning and a contextual understanding. Second, the benefit of rationales depends on the dataset used. While rationales significantly improve performance on RUStance-2023 and PStance, the gains are smaller or inconsistent on SemEval and GWStance, where stances are often expressed more explicitly in the text. This suggests that deliberative reasoning provides the greatest benefit for complex or ambiguous stance detection tasks rather than straightforward classification problems.

Across all datasets, the Fine-tuned LLM Judge trained on aggregated jury rationales achieves the best overall performance, achieving the highest Macro-F1 on three out of four datasets and the highest average

Macro-F1. Few-shot learning with rationales consistently outperformed zero-shot prompting, and both outperformed classical MLP baselines, indicating that jury-generated rationales provide useful task-specific reasoning signals. Overall, the results demonstrate that multi-LLM deliberation not only improves in-domain performance but also generalizes across multiple stance detection datasets, supporting the effectiveness of the proposed jury-based model.

5.2 RUStance-2023 Results

The detailed in-domain performance on the RUStance-2023 dataset (Table 4) across various model configurations reveals a clear and consistent improvement with additional deliberative information and task adaptation. The classical MLP baseline achieved the lowest performance, with a Macro-F1 of 0.428 using text only, which increased to 0.475 when jury rationales were included, indicating that deliberation text contains useful semantic information even for traditional machine learning models. For LLM-based approaches, zero-shot prompting with text only achieved a Macro-F1 of 0.454. When Round A rationales are added, the performance increases substantially to 0.543, and further improves to a Macro-F1 of 0.611 when both Round A and Round B rationales are included. This demonstrates that the deliberation process provides additional reasoning signals that help the model better interpret stances in ambiguous or context-dependent posts. Few-shot prompting with deliberation rationales further improves performance to a Macro-F1 of 0.647, showing that combining rationales with in-context examples provides complementary benefits.

The best performance is achieved by the fine-tuned LLM Judge, which reaches a Macro-F1 score of 0.791. This suggests that aggregated jury rationales are particularly effective as supervision signals for fine-tuning, enabling the Judge model to learn how expert annotations relate to collective jury reasoning and effectively bridge the gap between raw model outputs and authoritative human labels. Overall, the stepwise improvement across these configurations indicates that jury deliberation rationales provide progressively more value as the model learns to interpret and utilize them.

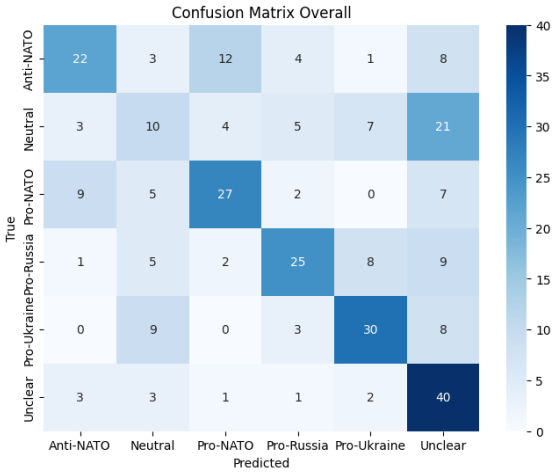
Table 4: Performance Comparison of Models on Stance Detection

Model	**Accuracy**	**Weighted F1**	**Macro-F1**
MLP (Text)	0.480	0.476	0.428
MLP (Text + Rationale AB)	0.513	0.510	0.475
Zero-shot LLM (Text)	0.471	0.454	0.454
Zero-shot LLM (Text + Rationale A)	0.546	0.543	0.543
Zero-shot LLM (Text + Rationale AB)	0.606	0.611	0.611
Few-shot LLM (Text + Rationale AB)	0.653	0.647	0.647
Finetuned LLM Judge (Text + Rationale AB)	**0.81**	**0.80**	**0.791**

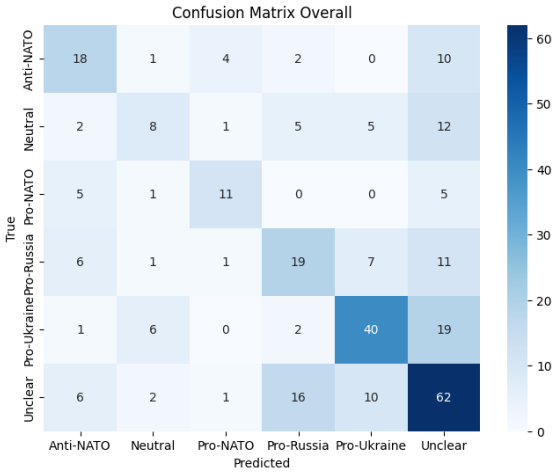
Figure 2 shows that performance improvements are not only reflected in overall Macro-F1, but also in clearer class-level separation. The MLP baselines show confusion, especially between the *Neutral* and *Unclear* classes, whereas the few-shot LLM settings produce a stronger diagonal structure, indicating better alignment with the stance schema. Adding the Round A and B rationales further reduced off-diagonal errors, particularly for the more clearly expressed stance categories. The fine-tuned Judge model achieved the strongest diagonal dominance overall, with the remaining errors concentrated mainly in the inherently ambiguous *Neutral* and *Unclear* classes. This pattern supports the view that jury rationales are especially valuable for resolving difficult cases while ambiguity remains concentrated in weakly signaled posts.

5.3 Cross-Domain Results

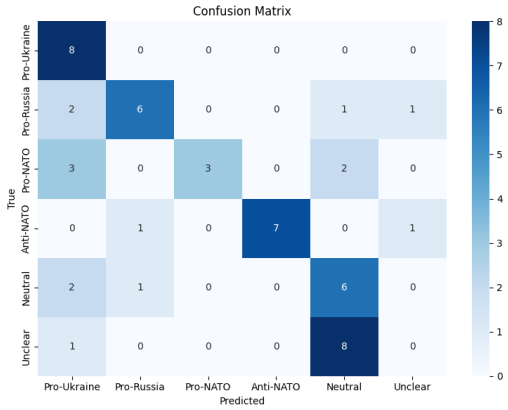
We evaluate our models on three publicly available stance detection datasets: PStance, SemEval-2016 Task 6, and GWStance. PStance is a political stance detection dataset consisting of over 21,000 tweets annotated with stances toward political figures, designed to support cross-target stance classification. SemEval-2016 Task 6 is a widely used benchmark dataset for stance detection on Twitter, where tweets are labeled as FAVOR, AGAINST, or NONE with respect to a given target. GWStance is a stance detection dataset



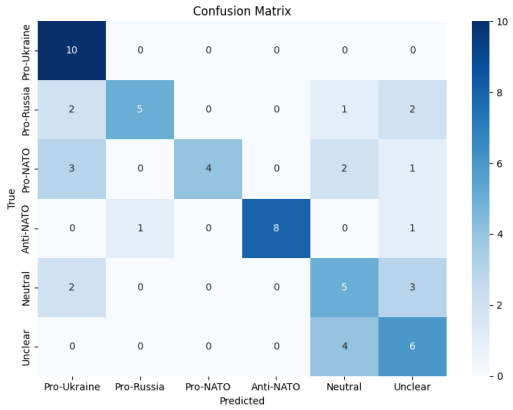
(a) MLP, Text Only



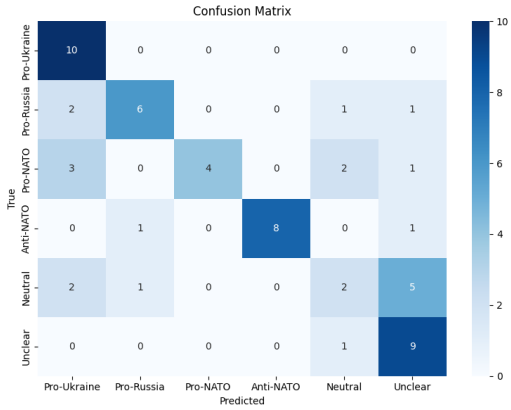
(b) MLP, Text and Rationale A & B



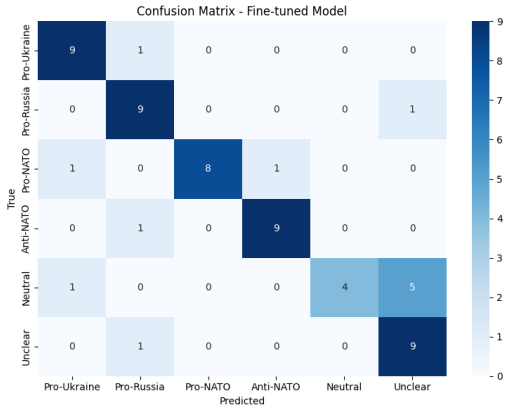
(c) Few-shot LLM, Text Only



(d) Few-shot LLM, Text combine Rationale A



(e) Few-shot LLM, Text combine Rationale A & B



(f) Fine-tuned Evaluation

Figure 2: Confusion matrices for the evaluated models.

derived from news articles related to global warming, in which sentences are annotated according to their stance toward climate change. To ensure a fair comparison across datasets and control for class imbalance, we randomly sampled 300 instances from each dataset and constructed a balanced subset with an equal

number of instances for each stance label. This balanced sampling strategy helps prevent model bias toward majority classes and allows for more reliable comparison of model performance across different datasets and experimental settings.

Table 5 shows model performance across three stance datasets. Incorporating rationales consistently improves the performance of text-only models. Few-shot and fine-tuned LLMs generally outperform MLP baselines, particularly on the PStance dataset, where the fine-tuned model achieves the best Macro-F1 score of 0.946. In SemEval-2016, the few-shot model performed best, suggesting that task-specific prompting may generalize better than fine-tuning on smaller datasets. On GWStance, both MLP with rationales and fine-tuned LLMs achieve strong performance, indicating that rationale-based reasoning benefits both traditional and LLM-based approaches.

Table 5: Model performance across three datasets: PStance, SemEval, and GWStance.

Model	**PStance**	**SemEval**	**GWStance**
MLP (Text)	0.675	0.628	0.885
MLP (Text + Rationale AB)	0.819	0.697	0.929
Zero-shot LLM (Text)	0.686	0.733	0.790
Zero-shot LLM (Text + Rationale AB)	0.922	0.705	0.790
Few-shot LLM (Text + Rationale AB)	0.943	0.736	0.783
Finetuned LLM Judge (Text + Rationale AB)	**0.946**	**0.693**	**0.911**

5.4 Statistical Significance Testing

To evaluate whether the performance differences between the model configurations were statistically reliable, we conducted paired bootstrap resampling on the Macro-F1 scores. The results are presented in Table 6. The analysis shows that few-shot prompting with jury rationales significantly outperforms zero-shot prompting with rationales ($\Delta F1 = 0.0367$, $p = 0.0026$), indicating that in-context examples and deliberative rationales provide complementary benefits to the model. Furthermore, the fine-tuned Judge model significantly outperforms the few-shot configuration ($\Delta F1 = 0.0636$, p= 0.0020), demonstrating that aggregated jury rationales serve as effective supervision signals for training a specialized stance classifier.

We observe that incorporating jury deliberation rationales significantly improves zero-shot prompting performance compared with text-only zero-shot prompting (Δ F1 = 0.1557, $p < 0.001$). This result indicates that deliberative rationales provide useful reasoning signals even without task-specific adaptation, enabling the model to better interpret ambiguous or context-dependent stance expressions. The improvement is consistent with the gains observed across the RUStance-2023 dataset, where rationale-enhanced prompting substantially outperforms text-only prompting. These findings suggest that jury-generated rationales provide complementary reasoning signals that improve stance classification performance. Deliberative rationales improve performance across all settings, with the largest gains obtained when combined with few-shot learning and fine-tuning Overall, the statistical significance analysis confirms that the improvements obtained from few-shot learning and fine-tuning with jury deliberation rationales are statistically significant and not due to random variations. These findings support the central hypothesis of this work, that structured multi-LLM deliberation provides useful supervisory signals that can improve stance classification when properly integrated into the learning process.

Table 6: Paired bootstrap comparison of model configurations using Macro-F1.

Comparison	Mean Diff	CI Lower	CI Upper	p-value
Few-shot vs Zero-shot (Text+Rationale AB)	0.0367	0.0115	0.0634	0.0026
Fine-tuned vs Few-shot (Text+Rationale AB)	0.0636	0.0178	0.1104	0.0020
Zero-shot (Text+Rationale AB) vs. Zero-shot (Text)	0.1557	0.0995	0.2124	< 0.001

Table 7: Performance comparison for 12-juror and 6-juror settings.

Jury	Model	F1	Acc	Input / Prompting
12	MLP	0.52	0.47	Text combine rationale A & B
12	Llama-3.3-8B Zero-shot	0.45	0.41	Text only
12	gpt-4.1-mini Zero-shot	0.51	0.51	Text only
12	Llama-3.3-8B Zero-shot	0.54	0.51	Text + rationale A
12	gpt-4.1-mini Zero-shot	0.58	0.58	Text + rationale A
12	Llama-3.3-8B Zero-shot	0.57	0.54	Text + rationale A &B
12	gpt-4.1-mini Zero-shot	0.60	0.60	Text + rationale A &B
12	Llama-3.3-8B Few-shot	0.60	0.59	Text combine rationale A & 3 examples
12	gpt-4.1-mini Few-shot	0.63	0.63	Text combine rationale A & 3 examples
12	Llama-3.3-8B Few-shot	0.65	0.65	Text combine rationale AB & 3 examples
12	gpt-4.1-mini Few-shot	0.65	0.63	Text combine rationale AB & 3 examples
12	gpt-4.1-mini Fine-tuned	0.81	0.80	Text + rationale A& B
6	MLP	0.42	0.42	Text combine rationale A &B
6	Llama-3.3-8B Zero-shot	0.43	0.41	Text only
6	gpt-4.1-mini Zero-shot	0.51	0.50	Text only
6	Llama-3.3-8B Zero-shot	0.56	0.54	Text combine rationale A
6	gpt-4.1-mini Zero-shot	0.53	0.51	Text combine rationale A
6	Llama-3.3-8B Zero-shot	0.51	0.51	Text combine rationale A & B
6	gpt-4.1-mini Zero-shot	0.56	0.56	Text combine rationale A & B
6	Llama-3.3-8B	0.63	0.62	Text combine rationale A & 3 examples
6	gpt-4.1-mini Few-shot	0.58	0.57	Text combine rationale A & 3 examples
6	Llama-3.3-8B Few-shot	0.53	0.53	Text combine rationale AB & 3 examples
6	gpt-4.1-mini Few-shot	0.63	0.62	Text combine rationale AB & 3 examples
6	gpt-4.1-mini Fine-tuned	0.69	0.62	Text combine rationale A & B

5.5 Ablation Study: Performance Comparison

12-Juror vs. 6-Juror Settings.

A cross-setting comparison revealed that reducing the jury from 12 to 6 members produced a consistent but moderate decline in performance across model classes, highlighting the value of juror diversity and rationale richness for downstream stance modeling. Classical baselines are most affected: the MLP drops from $(\text{F1} = 0.52, \text{Acc} = 0.47)$ in the 12-juror setting to $(0.42, 0.42)$ with only six jurors, reflecting the reduced coverage and semantic variability of the rationale embeddings. Zero-shot LLMs show similar sensitivity: Llama-3.3-8B decreases from $(0.45, 0.41)$ to $(0.43, 0.41)$ under text-only prompting, while models conditioned on concatenated Round A or Round A&B rationales exhibit small but systematic reductions of 0.02–0.04 F1. Few-shot prompting mitigates some of this degradation, but still exhibits a measurable gap. For example, the Llama-3.3-8B few-shot performance declines from (0.60–0.65) F1 under the 12-juror configuration to (0.58–0.63) when only six jurors are available, mirroring a similar contraction for gpt-4.1-mini.

The most notable trend is that while fine-tuned GPT-4.1-mini remains the top-performing model in both settings, its performance also reflects the reduction in available jury signals, decreasing from $(\text{F1} = 0.81, \text{Acc} = 0.80)$ with 12 jurors to $(0.69, 0.62)$ in the 6-juror setting. This demonstrates that although fine-tuning is robust to noise and model class variability, it still benefits substantially from the richer deliberation space afforded by a larger LLM jury. Overall, the comparison indicates that (i) more jurors produce more stable and higher-quality rationale embeddings; (ii) few-shot and fine-tuned models leverage these gains most effectively; and (iii) the performance separation between 12- and 6-juror settings widens with model sophistication, underscoring the importance of jury scale for high-fidelity stance annotation and model alignment.

Table 8: Performance of self-consistency with different numbers of samples on the RUStance-2023 dataset.

Method	Accuracy	Macro-F1	Weighted F1
Self-Consistency (K=3)	0.5667	0.5513	0.5513
Self-Consistency (K=12)	0.5900	0.5772	0.5772

Self-Consistency: K=3 vs. K=12 Setting

To compare the proposed multi-LLM jury model with a single-model ensemble strategy, we implemented a self-consistency baseline (Wang et al., 2023), where multiple stochastic outputs are sampled from the same language model and aggregated via majority voting. We used a single LLM (GPT-4o-mini) and generated independent predictions for each input using a temperature of 0.7 and top-p of 0.95 to encourage output diversity. This process was repeated for $K = 3$ and $K = 12$ samples per input, allowing us to compare the sampling diversity (self-consistency) with the model diversity (multi-LLM jury). For each tweet, the model was instructed to output exactly one stance label, and ties were resolved using a deterministic rule by selecting the earliest sampled label among the tied candidates.

The results on the RUStance-2023 dataset, shown in Table 8, indicate that increasing the number of samples improves performance, with Macro-F1 increasing from 0.551 to 0.577 as K increases from 3 to 12. This suggests that aggregating multiple stochastic outputs helps to stabilize predictions and allows the model to explore multiple reasoning paths before making a final decision. However, the performance gains from self-consistency remain smaller than those obtained from the multi-LLM jury model. While self-consistency introduces diversity through stochastic decoding of a single model, the jury model introduces diversity through multiple heterogeneous models with different architectures, training data, and reasoning patterns.

This comparison highlights an important distinction between sampling diversity and model diversity. Self-consistency primarily improves performance by reducing the variance in individual model outputs, whereas the proposed jury model leverages systematic disagreement between different models as a signal of ambiguity and uncertainty. The results suggest that model diversity provides a stronger signal than sampling diversity alone, particularly for complex stance detection tasks where subjective interpretation plays a significant role.

5.6 Failure Analysis

To better understand the limitations of the proposed jury Model, we conducted a qualitative analysis of misclassified examples. We observed that most model errors fell into a small number of recurring categories, such as implicit stance, sarcasm/irony, mixed stance, target confusion, ambiguity boundary, and context dependence, as summarized in Table 9. Detailed examples for each category are provided in Appendix A.

The most common errors occurred in tweets with implicit stance and sarcasm, where the stance is not explicitly stated and requires contextual interpretation. We also observed frequent errors in multi-target tweets, where the models disagreed on which political actor was the primary stance target. These findings suggest that many classification errors are caused by genuine linguistic ambiguity rather than simple model mistakes, which further supports the central premise of this work that disagreement between models is a signal of uncertainty in subjective NLP tasks.

6 Conclusion

This study proposes a deliberative *jury-of-LLMs* model for stance detection inspired by *12 Angry Men*. Unlike conventional annotation methods that reduce disagreement to a single consensus label, our approach explicitly preserves divergence through entropy, minority reports, and hung-jury flags. An empirical evaluation of the RUStance-2023 dataset demonstrated that supervised fine-tuning of gpt-4.1-mini achieved the strongest single-model performance (≈ 0.81 accuracy), and the jury model provided competitive accuracy with richer interpretability. By comparing human expert annotations, individual LLM baselines, and multi-LLM aggregation, we showed that disagreement can serve as a signal rather than noise, offering diagnostic insights into the uncertainty and subjectivity.

References

Miriam Louise Carnot, Lorenz Heinemann, Jan Braker, Tobias Schreieder, Johannes Kiesel, Maik Fröbe, Martin Potthast, and Benno Stein. On stance detection in image retrieval for argumentation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Taipei, Taiwan, 2023. ACM. doi: 10.1145/3539618.3591917.

Hyeong Kyu Choi, Xiaojin Zhu, and Yixuan Li. Debate or vote: Which yields better decisions in multi-agent large language models?, 2025. URL https://arxiv.org/abs/2508.17536.

Costanza Conforti, Mohammad Taher Pilehvar, and Nigel Collier. Towards automatic fake news detection: Cross-level stance detection in news articles. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pp. 78–83. Association for Computational Linguistics, 2018. URL https://aclanthology.org/W18-5507/.

Costanza Conforti, Mohammad Taher Pilehvar, and Nigel Collier. Stander: An expert-annotated dataset for news stance detection and evidence retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3256–3267, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.292.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28, 1979. doi: 10.2307/2346806. URL https://doi.org/10.2307/2346806.

Eve Fleisig, Rediet Abebe, and Dan Klein. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=ayzVnzaUzB.

Fabrizio Gilardi, Mohammad Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences of the United States of America*, 120 (30):e2305016120, July 2023. doi: 10.1073/pnas.2305016120. URL https://doi.org/10.1073/pnas.2305016120.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. Stance detection in COVID-19 tweets. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1596–1611, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.127. URL https://aclanthology.org/2021.acl-long.127/.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. Semeval-2019 task 7: Rumoureval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*, pp. 845–854, Minneapolis, Minnesota, USA, 2019. Association for Computational Linguistics. doi: 10.18653/v1/S19-2147.

Sulbha Jain. Replacing llm judge with juries: Example, 4 2025. URL https://medium.com/@sulbha.jindal/replacing-llm-judge-with-juries-example-97bb63d4a376.

Anton Lavrouk, Ian Ligon, Jonathan Zheng, Tarek Naous, Wei Xu, and Alan Ritter. Stanceosaurus 2.0 - classifying stance towards Russian and Spanish misinformation. In Rob van der Goot, JinYeong Bak, Max Müller-Eberstein, Wei Xu, Alan Ritter, and Tim Baldwin (eds.), *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pp. 31–43, San Ġiljan, Malta, March 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.wnut-1.4/.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. P-stance: A large dataset for stance detection in political domain. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2355–2365, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.208. URL https://aclanthology.org/2021.findings-acl.208/.

Yue Li, Jake Vasilakes, Zhixue Zhao, and Carolina Scarton. Scrum-9: Multilingual stance classification over rumours on social media, 2025. URL https://arxiv.org/abs/2505.18916.

Yiwei Luo, Dallas Card, and Dan Jurafsky. Detecting stance in media on global warming. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3296–3315, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.296. URL https://aclanthology.org/2020.findings-emnlp.296/.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. SemEval-2016 task 6: Detecting stance in tweets. In Steven Bethard, Marine Carpuat, Daniel Cer, David Jurgens, Preslav Nakov, and Torsten Zesch (eds.), *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 31–41, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/S16-1003. URL https://aclanthology.org/S16-1003/.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110, 2022. doi: 10.1162/tacl_a_00449. URL https://aclanthology.org/2022.tacl-1.6/.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. Stance detection benchmark: How robust is your stance detection? *KI - Künstliche Intelligenz*, 35(3):329–341, November 2021. ISSN 1610-1987. doi: 10.1007/s13218-021-00714-w. URL https://doi.org/10.1007/s13218-021-00714-w.

Yiyao Tao, Hengyu Zhang, Babli Dey, Selenge Tulga, Hanjia Lyu, and Jiebo Luo. In the Eyes of the Bystander: Are the Stances on Different Conflicts Correlated? . In *2024 IEEE International Conference on Big Data (BigData)*, pp. 7193–7201, Los Alamitos, CA, USA, December 2024. IEEE Computer Society. doi: 10.1109/BigData62323.2024.10825370. URL https://doi.ieeecomputersociety.org/10.1109/BigData62323.2024.10825370.

Alexandra Uma, Tommaso Fornaciari, Silviu O. Paun, Barbara Plank, and Massimo Poesio. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470, 2021. doi: 10.1613/jair.1.12108.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Xiutian Zhao, Ke Wang, and Wei Peng. An electoral approach to diversify LLM-based multi-agent collective decision-making. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2712–2727, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.158. URL https://aclanthology.org/2024.emnlp-main.158/.

Appendix A: Qualitative Analysis of Model Errors

This section presents a qualitative analysis of common error patterns observed in stance classification. We grouped the errors into several recurring categories to better understand the limitations of current LLM-based stance detection systems.

Table 9: Common failure categories observed in stance classification.

Failure Type	Description
Implicit stance	Stance implied without explicit keywords.
Sarcasm/irony	Literal wording opposite to the intended stance.
Mixed stance	Multiple political targets in one tweet.
Target confusion	Incorrect identification of the stance target.
Ambiguity boundary	Neutral statements interpreted as unclear.
Context dependence	Requires external knowledge.

A.1 Ambiguity Boundary Failures

Case 1 (False Ambiguity). *Tweet:* "Parliament will debate the new trade bill tomorrow." *Expert label:* Neutral; *LLM output:* Unclear (all models). *Explanation:* The models interpreted the absence of emotional or ideological cues as uncertainty, even though the tweet is a purely factual political update.

Case 2 (Keyword Over-Sensitivity). *Tweet:* 'Weather in Kyiv looks calm today despite tensions."
Expert label: Neutral; *LLM output:* Unclear (GPT-3.5, Gemini-2.5).
Explanation: The word 'tensions" triggered a stance inference, although it is used descriptively and contains no evaluative or political positioning.

Case 3 (Vagueness Misinterpreted as Uncertainty). *Tweet:* "Can't believe how politics always distracts from real issues." *Expert label:* Neutral; *LLM output:* Unclear (GPT-4o-mini). *Explanation:* The model treated general political commentary as ambiguity, whereas expert annotators interpreted it as politically disengaged neutral content.

A.2 Subtle Context or Narrative Failures

Case 1 (Evaluative Framing Ignored). *Tweet:* "How Putin's dream of a new Russia turned into a nightmare for his own people." *Expert label:* Pro-Ukraine; *LLM output:* Unclear (GPT-4o-mini). *Explanation:* The negative evaluation of Putin implies a pro-Ukraine stance, but the model interpreted the sentence as neutral reporting.

Case 2 (Metaphorical Critique Missed). *Tweet:* 'Europe's challenge now is to deal with Russia's imperial illusions."
Expert label: Pro-NATO; *LLM output:* Unclear (GPT-3.5).
Explanation: The metaphorical phrase 'imperial illusions" conveys ideological criticism of Russian expansionism, which the model failed to interpret as pro-Western alignment.

Case 3 (Abstract Moral Language Discounted). *Tweet:* "We Are All One Family — Ukraine NATO Freedom." *Expert label:* Pro-NATO; *LLM output:* Unclear (GPT-.1). *Explanation:* Although the hashtags signal Western alignment, the abstract moral phrasing led the model to interpret the stance as ambiguous.

A.3 Target Confusion

Case 1 (Sarcastic Framing Misinterpreted). *Tweet:* 'Unprovoked aggression??? Russia"
Expert label: Pro-Russia; *LLM output:* Pro-Ukraine (GPT-3.5).
Explanation: The model focused on the word 'aggression" and assumed criticism of Russia, missing the sarcastic quotation framing.

Case 2 (Multi-Target Stance Conflict). *Tweet:* "@NATO you failed to protect humanity. Like Iraq, like Libya, now Ukraine." *Expert label:* Anti-NATO; *LLM output:* Pro-Ukraine (GPT-3.5, Gemini-2.5). *Explanation:* The tweet criticizes NATO while expressing concern about Ukraine. Expert annotators labeled the tweets based on the primary target of criticism (NATO), whereas the LLMs focused on the sympathetic reference to Ukraine.

Case 3 (Alliance Keyword Confusion). *Tweet:* 'Joe Biden's America will always get worse. Same playbook, same allies."
Expert label: Anti-NATO; *LLM output:* Pro-Ukraine (all models).
Explanation: The word 'allies" was interpreted as a positive NATO reference, although it was used sarcastically to criticize Western leadership.