More Modalities, More Problems: When Multimodal Agentic Debate Hurts Time-Series Reasoning

Patara Trirat
DeepAuto.ai
patara@deepauto.ai

Abstract

Multi-agent debate is often motivated by the premise that multiple agents, modalities, and rounds of deliberation improve reasoning by combining complementary evidence and correcting individual errors. We test this premise in multimodal time-series analysis using **MAD-TS**, a prompting-based framework with modality-specific agents, cross-modal judging, and synthesis. Contrary to expectation, MAD-TS does not reliably outperform simpler baselines. Across 20 tasks in finance, weather, healthcare, and science, it is uniquely best on 2 of 23 outcomes and significantly beats a strong single-agent multimodal baseline on only one task, while requiring about 22× more tokens and 12× more wall-clock time. Adding modalities, increasing debate depth, or changing judge rounds can help some tasks but harm others. Trace analysis attributes 85% of errors to all three judges agreeing on the same wrong answer, a shared upstream bias that limits same-backbone multimodal debate as a general-purpose approach.

1 Introduction

Background A common design intuition is that multimodal inputs and multi-agent debate should improve reliability. Different modalities provide complementary evidence, and debate can surface and correct mistakes. Prior work supports these ingredients separately. Debate improves reasoning in text-only settings (Du et al., 2024; Liang et al., 2024), and vision–language models effectively integrate multimodal inputs (Gemini Team et al., 2024). This motivates the hypothesis that *debate over multimodal evidence should outperform simpler single-model or single-modality baselines*.

Related Work Most evidence for the benefits of multi-agent debate comes from language-only benchmarks with controlled setups (Du et al., 2024; Liang et al., 2024). Whether these gains transfer to settings that require integrating heterogeneous evidence remains unclear. Closest to our study, Zhang et al. (2025) show across five MAD methods, nine benchmarks, and four models that homogeneous debate rarely outperforms strong single-agent baselines and that model heterogeneity is the operative lever. We test the complementary lever, heterogeneity of modal evidence under a fixed backbone, and find it insufficient to decorrelate same-backbone judges or produce reliable gains. Meanwhile, strong multimodal models such as Gemini (Gemini Team et al., 2024) and GPT-4o (OpenAI, 2024) achieve high performance on many tasks, yet their behavior under multi-agent interaction with potentially conflicting cross-modal evidence is not well characterized. Recent work has also highlighted brittle or misleading forms of deliberative reasoning and reliability issues in LLM reasoning (Shojaee et al., 2025; Mirzadeh et al., 2025; Zhou et al., 2024). Our study con-

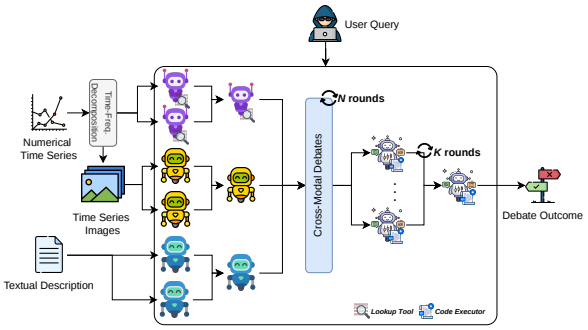


Figure 1: Overview of MAD-TS. Modality-specific agents first reason over individual inputs, judges then perform cross-modal deliberation, and a final synthesizer produces the task prediction.

Modality	Input	Hypothesized Strength
Text	News, reports	Context, sentiment, forward-looking indicators
Visual	Time series charts	Patterns, trends, visual anomalies
Numerical	Raw values, statistics	Precise calculations, quantitative features
Frequency	Spectral plots	Periodicity, dominant cycles

Table 1: Modalities used in MAD-TS, with representative inputs and their expected complementary strengths.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 945–957
November 6-10, 2026 ©2026 Association for Computational Linguistics

Task	MAD-TS	ByMyEyes	MAD	MM-MAD	SA-CoT	MM-SA-CoT	SA	MM-SA	VLTime
				MTBench (Chen et al., 2025)					
Finance Trend	0.427	**0.473**	0.347	0.457	0.310	0.437	0.317	0.437	0.290
Finance Forecasting (↓)	2.80	**1.59**	1.79	1.77	1.79	**1.59**	1.81	1.61	1.79
Finance Indicator (↓)	6.66	0.21	**0.20**	8.13	0.21	4.00	0.21	0.34	0.49
Finance Correlation	0.293	**0.317**	0.267	0.250	0.240	0.303	0.227	0.313	0.253
Finance MCQA	0.803	**0.873**	0.780	0.817	0.797	0.817	0.823	0.847	0.727
Weather Trend	0.310	0.450	0.223	0.320	0.220	0.477	0.250	**0.523**	0.247
Weather Forecasting (↓)	4.21	3.68	4.00	3.86	4.09	**3.52**	3.92	3.53	4.66
Weather Indicator, diff (↓)	5.26	4.67	4.81	4.27	3.53	4.68	5.11	4.50	**3.39**
Weather Indicator, max (↓)	4.82	4.44	4.69	3.58	**3.51**	4.34	5.24	4.30	3.58
Weather Indicator, min (↓)	3.52	2.99	**2.82**	2.90	3.03	2.89	2.94	2.90	3.09
Weather MCQA	0.593	0.637	0.503	0.480	0.600	0.617	0.617	**0.653**	0.580
				TimerBed (Liu et al., 2025b)					
ECG	0.250	0.243	0.150	0.237	0.187	0.247	0.220	0.250	**0.257**
HAR	0.288	0.198	0.257	0.229	0.295	**0.312**	0.253	0.233	0.306
EMG	0.236	0.301	0.195	0.285	**0.455**	0.203	0.154	0.236	0.171
CTU	**0.537**	0.477	0.523	0.510	0.483	0.500	0.507	0.503	0.500
RCW	0.470	**0.507**	0.497	0.483	0.497	0.477	0.470	0.483	0.453
TEE	0.192	0.160	0.132	0.169	0.137	**0.247**	0.128	0.187	0.123
				TSQA (Kong et al., 2025a)					
Anomaly Detection	0.543	0.547	0.580	0.503	**0.583**	0.533	0.543	**0.583**	0.523
Classification	**0.640**	0.277	0.553	0.557	0.510	0.513	0.213	0.243	0.443
Forecasting (↓)	111.24	97.46	120.13	153.52	105.36	91.68	95.93	**88.32**	103.19
Imputation (↓)	24.97	**6.66**	15.37	13.66	33.57	11.06	96.85	6.73	8.65
QA, true/false†	0.58	0.75	0.80	0.78	**0.83**	0.71	0.77	0.65	0.52
QA, multiple-choice†	0.35	0.53	**0.61**	0.55	0.36	0.32	0.41	0.54	0.35

Table 2: Main empirical results across MTBench, TimerBed, and TSQA benchmark suites. All cells are 3-run means (100 samples per run). Accuracy ↑ unless marked MAE ↓. Bold marks the best in each row. †QA rows cover the true/false and multiple-choice subtypes only. Standard deviations across runs and significance details are reported in §E.

nects these threads by evaluating debate in a multimodal setting, showing that additional interaction and complexity can amplify failure modes rather than mitigate them, consistent with emerging concerns about risks arising from interaction dynamics in agentic systems (Lynch et al., 2025).

Approach and Contributions We study whether multimodal agentic debate improves time-series reasoning by evaluating MAD-TS across 20 tasks spanning finance, weather, healthcare, and scientific domains under a fixed backbone. Contrary to common expectations (Kong et al., 2025b; Liu et al., 2025a), we observe consistent negative patterns: combining modalities often underperforms the best single modality, additional debate rounds do not reliably help, and prompt refinements produce non-monotonic trade-offs. We make three contributions. First, we show that multimodal debate does not reliably beat strong single-agent multimodal baselines despite ∼22× more tokens, a result that is stable across three runs and two backbones. Second, we attribute 85% of errors over 2,688 trajectories to shared modality-agent bias and at most ∼5% to strict majority-over-quality (Table 5). Third, we report a qualitative negative result on 12 prompt-level mitigations, which indicates that architectural or training-time interventions are likely required.

Method	Tokens/Sample	Time/Sample	Total Tokens	Total Time
MAD-TS	136,678	55.0s	261.1M	29.2h
MM-SA	6,341	4.7s	12.1M	2.5h
Ratio	-	-	**21.6**×	**11.7**×

Table 3: Compute comparison between MAD-TS and a strong MM-SA baseline.

2 Problem and Experimental Setups

Problem Formulation As summarized in Table 1, we study multimodal time-series tasks where each instance may include complementary views of the same signal: textual context (**T**), visual plots (**V**), numerical features (**N**), and frequency-domain representations (**F**). The objective is to predict task-specific outputs, including class labels, multiple-choice answers, or real-valued forecasts. The benchmark spans 20 tasks across four domains, covering both classification and regression settings.

Debate Architecture MAD-TS assigns three role-differentiated agents (proposer, challenger, analyst) to each available modality. Each agent first produces modality-specific reasoning, which is then compared and integrated by a panel of judges through cross-modal deliberation. A final synthesizer aggregates the judges' outputs into a single prediction. The design is motivated by two hypotheses: modality-specialized agents capture complementary evidence, and deliberation across judges improves reliability by reconciling conflicting signals. As illustrated in Figure 1, the framework operates in three stages.

Debate Mechanics In each debate round $n \in \{1, \ldots, N\}$, every modality agent a_m receives its input x_m, the current-round outputs of the earlier roles in its modality, and the previous-round outputs of all other agents. It is then prompted to defend, refine, or revise its position. After N agent rounds, J judges deliberate over the agents' final outputs for K judge rounds. The synthesizer is a single LLM call over the judges' final outputs that emits the task prediction with no further deliberation (Algorithm 1, prompts in §A.5). We use $N{=}2$, $J{=}3$, $K{=}1$ (§A.3).

Baselines We compare MAD-TS against simpler baselines covering single-agent text-only and multimodal systems, chain-of-thought variants, and text-only or multimodal debate setups. This comparison isolates whether the added complexity of MAD-TS yields gains beyond standard multimodal prompting and simpler debate-based approaches.

- **Single Agent (SA)**: Text-only baseline.
- **SA-CoT**: Text-only baseline with chain-of-thought (CoT) reasoning (Wei et al., 2022).
- **MM-SA**: Multimodal (MM) variant of SA.
- **MM-SA-CoT**: Multimodal SA with CoT.
- **Multi-Agent Debate (MAD)**: Text-only multi-agent debate baseline (Liang et al., 2024).
- **MM-MAD**: Multimodal variant of MAD.
- **VLTime**: Vision-language single-agent baseline for time series (Liu et al., 2025b).
- **ByMyEyes**: Visual reasoning single-agent baseline for time series (Yoon et al., 2024).

Evaluation Protocol All methods use the same underlying model, `gpt-4.1-mini`, to ensure that differences are attributable to prompting and system design rather than model capacity. Main comparisons aggregate results over three runs with 100 samples per task. Every method sees the same examples in each run. Ablation studies use a smaller setting with 20 samples per task. We report accuracy for classification-style tasks and mean absolute error for regression-style tasks. In addition to predictive performance, we analyze computational cost in tokens and wall-clock time, since a central claim of debate frameworks is that extra computation should buy better reasoning.

3 Empirical Studies

Main Results Table 2 shows that **MAD-TS does not provide reliable gains over much simpler baselines**. Across 23 outcomes on 20 tasks, it ranks first only twice, while MM-SA and MM-SA-CoT remain competitive on most tasks. Paired tests over three runs confirm the pattern. We compare MAD-TS with MM-SA on the 13 classification/MCQA tasks using exact McNemar tests with Holm correction. MAD-TS is significantly better on a single task, TSQA Classification, in every run, and on no other task in any run. On the four regression tasks, its MAE is worse in all four, and every run-level bootstrap interval that excludes zero favors MM-SA (§E). Table 3 shows a poor cost-performance trade-off. MAD-TS uses 22× more tokens and 12× more inference time than MM-SA without commensurate gains. The picture is unchanged under Gemini 2.5 Flash, where the single significant win is a different task (§F). **More agents and more deliberation do not inherently improve multimodal time-series reasoning.**

Modality Ablations To test whether broad multimodal input is itself the bottleneck, we evaluate all 15 modality combinations on the finance tasks (20 samples per task). Figure 2 shows that performance is strongly task-dependent. F is best for trend prediction, NV for price forecasting, V for indicator forecasting, and TN/TF for MCQA. Crucially, the full TNVF configuration is *not* best on any metric. Within this ablation, adding modalities does not produce monotonic gains. Instead, it often introduces trade-offs and possible modality interference. The benefit of multimodality appears conditional on task-specific signal alignment rather than simple input breadth.

Component Ablations Table 4 shows that, on the finance tasks studied, MAD-TS components do not contribute reliably. Tool access is not consistently beneficial. Removing lookup sometimes improves forecasting. Judges help on

Ablation	Configuration	Trend ↑	Forecast ↓	Indicator ↓	Correlation ↑	MCQA ↑
Judge tools	Lookup only	0.40	1.55	**36.41**	**0.40**	**0.90**
	Code only	**0.50**	1.42	72.69	0.30	0.85
	Neither	0.45	**1.34**	39.29	**0.40**	0.85
Judge modules	Only judges	**0.45**	**1.44**	**28.66**	**0.50**	–
	w/o judges	**0.45**	1.46	49.82	0.20	–
Agent-side tools	w/o numerical lookup	**0.30**	1.30	**0.20**	0.45	–
	w/o numerical judge lookup	0.20	1.20	0.25	**0.50**	–
	w/o both	**0.30**	**1.15**	0.59	0.45	–

Table 4: Component ablations of MAD-TS on the finance tasks (20 samples per task). Bold marks the best configuration within each ablation. The judge-module and agent-side ablations did not include MCQA.

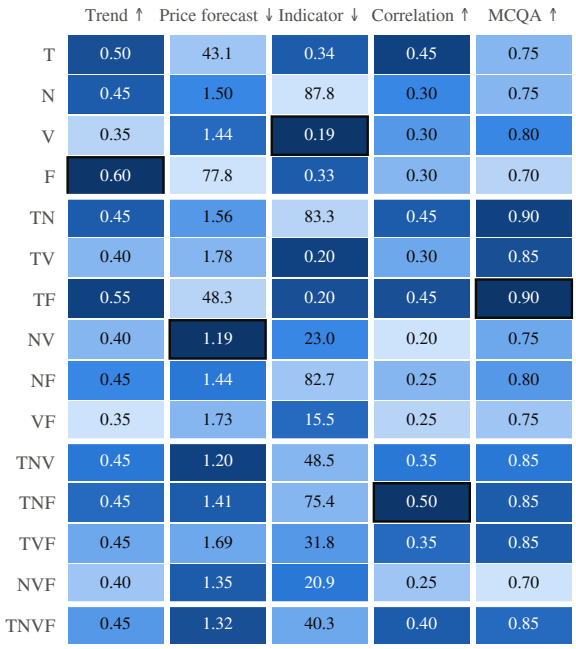


Figure 2: Modality ablation on the five finance tasks (20 samples per task). Rows are modality combinations (T text, N numerical, V visual, F frequency). Darker shading marks a better rank within each column, and the outlined cell is the best configuration for the task. The full configuration TNVF is best on no task.

indicator and correlation but the gains do not transfer broadly. MAD-TS instantiates a hierarchical topology (modality agents → judges → synthesizer). The judge-module rows of Table 4 implicitly contrast it with flatter variants. "Only judges" approximates a flat panel and "w/o judges" a round-table debate, and neither uniformly improves over the full hierarchy. A sweep over alternative topologies (star, ring, blackboard) is left to future work.

Hyperparameter Sensitivity Varying debate depth and judge depth (20 samples per task) yields similarly unstable behavior. More judge rounds (Figure 3, left) improve some healthcare tasks but hurt finance trend and forecasting. More debate rounds (Figure 3, right) reduce some regression errors while degrading classification accuracy. Varying the number of judges on finance tasks gives the same flat, non-monotonic profile. The preferred setting changes by task and the direction of the effect reverses across domains, so the framework requires task-specific tuning despite its high inference cost. These 20-sample cells are noisy (median std 0.050, max 0.125, §E), so we read Figures 2 and 3 and Table 4 as evidence of task dependence rather than of suite-wide component effects.

Failure regime (of 1,447 errors)	Count	%
All 3 judges agreed on the same wrong answer	1,230	**85.0%**
Judges disagreed	161	11.1%
At least one judge had the correct answer	77	5.3%
Strict majority-over-quality (upper bound)	≤77	≤5.3%
Judges unparsable	56	3.9%

Table 5: Failure attribution on 9 classification/MCQA tasks (2,688 samples over three runs, 1,447 errors). Cases where all judges share the same wrong answer dominate cases where judges disagree. The breakdown by task is in Table 18.

Failure Patterns We parse each MAD-TS trace into the judge answers and the synthesizer's final answer on the nine classification/MCQA tasks (Table 5, 2,688 samples over three runs, 1,447 errors). In 85% of errors, all three judges agreed on the same wrong answer. This is evidence of *shared modality-agent bias*, not of aggregation failure. Only 11% of errors involved judge disagreement, and at most ∼5% fit the strict majority-over-quality template. Part of this agreement is mechanical, because the judges share one backbone and decode

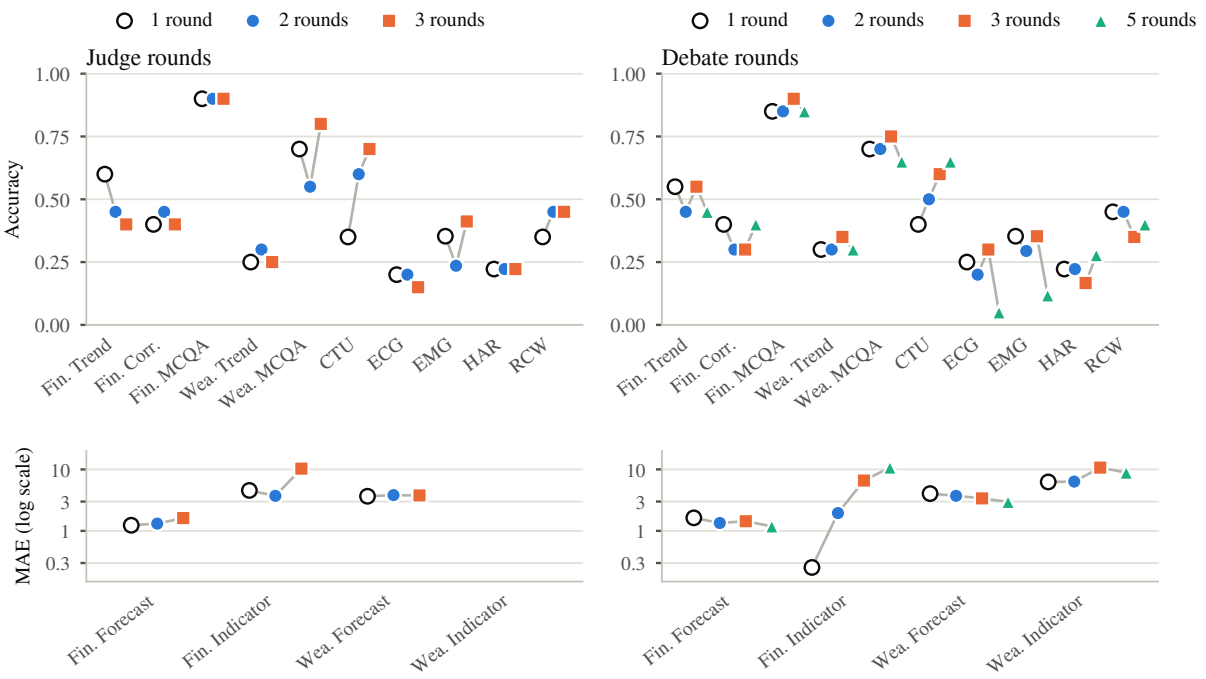


Figure 3: Sensitivity to judge depth (left) and debate depth (right) at 20 samples per task. Each marker is one setting, and the gray line joins the settings of a task in order. The top row shows accuracy for the classification and MCQA tasks, and the bottom row shows MAE for the regression tasks on a log scale. No setting is best on every task.

greedily, and both backbones we test are homogeneous (96% unanimity under Gemini 2.5 Flash, §F). We therefore read the rate as an observation about same-backbone debate rather than as proof of its cause. Qualitatively, traces still exhibit *vote-counting*, *temporal confusion*, *confidence asymmetry*, and *evidence abandonment*. These mechanisms have analogues in text-only debate. The dominant locus, however, is shared upstream signal. Prompt-level mitigations underperform because agents and judges from the same backbone share the same biases. An exploratory taxonomy is given in §G.

4 Discussion and Conclusions

In multimodal time-series analysis, a same-backbone multimodal agentic debate framework does not reliably outperform simpler baselines across 20 tasks despite requiring $\sim$22$\times$ more compute. **MAD-TS** is uniquely best on 2 of 23 outcomes and significantly beats a strong single-agent baseline on one task per backbone. It shows inconsistent effects across modalities, debate rounds, and judging depth in targeted ablations, and it exhibits 15 recurring failure modes. None of the 12 prompt interventions we tried produced consistent gains. The dominant failure mode is *shared upstream bias* rather than a good minority view being overruled by the synthesizer. In 85% of errors, all three judges agree on the same wrong answer (Table 5).

These results define practical limits for prompting-based, same-backbone multimodal agentic debate. We recommend benchmarking against strong simple baselines, starting with the best single modality, testing minimal deliberation, and evaluating across the full task suite. Productive future directions are architectural. They include learned aggregators that score reasoning quality instead of counting votes, verifier models trained to detect temporal confusion and false precision, adaptive routing on task type, and modality-asymmetric agents (specialist backbones per modality) whose disagreement is informative enough for the synthesizer to arbitrate.

Limitations

Our findings should be interpreted in light of several limitations. First, the main comparison uses a single underlying model (`gpt-4.1-mini`) so that differences are attributable to system design rather than model capacity. The Gemini 2.5 Flash replication in §F covers MAD-TS and MM-SA only, and a heterogeneous-backbone MAD-TS is untested, so the shared-bias explanation is an observation rather than a causal demonstration. Second, all methods decode greedily (temperature 0), which removes sampling diversity and makes part of the

judge agreement mechanical, so our conclusions do not extend to designs that sample for diversity. Third, the conclusions are scoped to prompting-based multimodal time-series reasoning and do not bound training-time or learned-aggregator approaches. Fourth, the failure attribution (Table 5) covers the 9 classification/MCQA tasks where judge answers parse cleanly, because judge outputs on regression and free-form QA are not categorical, and the failure taxonomy (§G) is exploratory. Finally, our analysis is diagnostic and the ablations are targeted (mostly finance, 20 samples per task). We document when and how debate fails, but we do not test alternative topologies, fine-tuning, or adaptive routing. We therefore view the paper not as a universal negative claim about agentic methods, but as evidence that their benefits in prompting-based multimodal time-series tasks are fragile, compute-intensive, and configuration-dependent in the setting we study.

Acknowledgments

This work was conducted as part of the research activities at DeepAuto.ai.

References

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. 2025. MTBench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *ICML*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, and 1118 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Yaxuan Kong, Yiyuan Yang, Yoontae Hwang, Wenjie Du, Stefan Zohren, Zhangyang Wang, Ming Jin, and Qingsong Wen. 2025a. Time-MQA: Time series multi-task question answering with context enhancement. In *ACL*, pages 29736–29753.

Yaxuan Kong, Yiyuan Yang, Shiyu Wang, Chenghao Liu, Yuxuan Liang, Ming Jin, Stefan Zohren, Dan Pei, Yan Liu, and Qingsong Wen. 2025b. Position: Empowering time series reasoning with multimodal llms. *Preprint*, arXiv:2502.01477.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *EMNLP*, pages 17889–17904.

Haoxin Liu, Harshavardhan Kamarthi, Zhiyuan Zhao, Shangqing Xu, Shiyu Wang, Qingsong Wen, Tom Hartvigsen, Fei Wang, and B. Aditya Prakash. 2025a. How can time series analysis benefit from multiple modalities? a survey and outlook. *Preprint*, arXiv:2503.11835.

Haoxin Liu, Chenghao Liu, and B. Aditya Prakash. 2025b. A picture is worth a thousand numbers: Enabling LLMs reason about time series via visualization. In *NAACL*, pages 7486–7518.

Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J Ritchie, Soren Mindermann, Evan Hubinger, Ethan Perez, and Kevin Troy. 2025. Agentic misalignment: How llms could be insider threats. *arXiv preprint arXiv:2510.05179*.

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *ICLR*.

OpenAI. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Parshin Shojaee, Seyed Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. In *NeurIPS*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Hyungjun Yoon, Biniyam Aschalew Tolera, Taesik Gong, Kimin Lee, and Sung-Ju Lee. 2024. By my eyes: Grounding multimodal large language models with sensor data via visual prompting. In *EMNLP*, pages 2219–2241.

Hangfan Zhang, Zhiyao Cui, Jianhao Chen, Xinrun Wang, Qiaosheng Zhang, Zhen Wang, Dinghao Wu, and Shuyue Hu. 2025. Stop overvaluing multi-agent debate–we must rethink evaluation and embrace model heterogeneity. *arXiv preprint arXiv:2502.08788*.

Lexin Zhou, Wout Schellaert, Fernando Martínez-Plumed, Yael Moros-Daval, Cèsar Ferri, and José Hernández-Orallo. 2024. Larger and more instructable language models become less reliable. *Nature*, 634(8032):61–68.

A Complete Experimental Setup

This section provides a comprehensive description of the experimental design, including task definitions, evaluation metrics, hyperparameter configurations, and ablation settings. These details are intended to facilitate reproducibility and clarify the scope of our evaluation.

A.1 Algorithm

Algorithm 1: MAD-TS Inference

Input: multimodal inputs $\{x_m\}_{m\in\mathcal{M}}$; debate rounds N; judges J; judge rounds K; roles $\mathcal{R} = \{\text{proposer, challenger, analyst}\}$

Output: prediction $\hat{y}$

// $A^{(n)}$ denotes all agent outputs of round n, with $A^{(0)}=\emptyset$

// $A^{(n)}_{m,\prec r}$ denotes the round-n outputs of modality m before role r

// $G^{(k)}_{-j}$ denotes the round-k outputs of the judges other than j

// Stage 1: Modality-specific agent debate

for $n = 1, \ldots, N$ **do**

  for $m \in \mathcal{M}$, $r \in \mathcal{R}$ in role order **do**

    $a^{(n)}_{m,r} \leftarrow \text{LLM}\big(\pi_{\text{agent}}(m, r, x_m, A^{(n)}_{m,\prec r}, A^{(n-1)})\big)$

// Stage 2: Cross-modal judge deliberation

for $j = 1, \ldots, J$ **do**

  $g^{(1)}_j \leftarrow \text{LLM}(\pi_{\text{judge}}(j, A^{(N)}))$

for $k = 2, \ldots, K$ **do**

  for $j = 1, \ldots, J$ **do**

    $g^{(k)}_j \leftarrow \text{LLM}\big(\pi_{\text{jrefine}}(j, A^{(N)}, G^{(k-1)}_{-j})\big)$

// Stage 3: Synthesis (single call, no deliberation)

$\hat{y} \leftarrow \text{LLM}(\pi_{\text{synth}}(\{g^{(K)}_j\}_{j=1}^{J}))$;

return $\hat{y}$

A.2 Task Details

Table 6 summarizes the full set of benchmark domains and tasks used in our evaluation. The benchmark spans multiple domains (MTBench-Finance, MTBench-Weather, TimerBed, and TSQA) and covers diverse task types, including classification, regression, multiple-choice question answering (MCQA), and open-ended QA.

Domain	Task	Type	Metric
Finance	Trend	Classification	Accuracy
Finance	Forecasting	Regression	MAE
Finance	Indicator (MACD)	Regression	MAE
Finance	Correlation	Classification	Accuracy
Finance	MCQA	MCQA	Accuracy
Weather	Trend	Classification	Accuracy
Weather	Forecasting	Regression	MAE
Weather	Indicator (diff, max, min)	Regression	MAE per component
Weather	MCQA	MCQA	Accuracy
TimerBed	ECG, HAR, EMG, CTU, RCW, TEE	Classification	Accuracy/F1
TSQA	Anomaly, Classification	Classification	Accuracy
TSQA	Forecasting, Imputation	Regression	MAE
TSQA	QA (true/false, multiple-choice)	QA	Accuracy

Table 6: Overview of benchmark domains, tasks, task types, and evaluation metrics.

Parameter	Default	Configuration Range
Agents per Modality	3	3
Max Debate Rounds	2	1, 2, 3, 5
Max Judge Rounds	1	1, 2, 3
Number of Judges	3	1, 3, 5

Table 7: Default hyperparameters and ranges used in our experiments.

Task Types Classification tasks require predicting discrete labels (e.g., trend direction or anomaly detection), and are evaluated using Accuracy (and F1 when class imbalance is present). Regression tasks involve predicting continuous values such as future time-series values or imputations, and are evaluated using Mean Absolute Error (MAE). MCQA tasks require selecting the correct option among candidates and are evaluated with Accuracy. QA tasks involve generating free-form answers. We score the true/false and multiple-choice subtypes by exact match. We do not report the open-ended subtype, which lacks a validated metric. The weather indicator task is scored per component (difference, maximum, minimum).

Domains Finance and Weather tasks focus on real-world temporal reasoning such as trend prediction and forecasting. TimerBed consists of physiological and sensor-based datasets (e.g., ECG, HAR, EMG), emphasizing signal classification. TSQA is a comprehensive time-series benchmark covering anomaly detection, classification, forecasting, imputation, and question answering.

Ablation	Configs
Modality	T, N, V, F, TN, TV, TF, NV, NF, VF, TNV, TNF, TVF, NVF, TNVF
Judge Tools	Code Only, Lookup Only, Neither
Agent Config	Only Judges, w/o Judges
Components	w/o Numerical Lookup, w/o Numerical Judge Lookup, w/o Both

Table 8: Configuration space for the ablation study.

A.3 Hyperparameters

Table 7 lists the key hyperparameters governing the multi-agent interaction process.

Max Rounds controls the number of interaction rounds among agents (e.g., debate steps).

Max Judge Rounds determines how many iterative evaluation or adjudication steps are performed by the judge agents.

Number of Judges specifies how many independent judge agents are used for aggregation or consensus.

A.4 Component Ablations

To understand the contribution of each system component, we conduct systematic ablation studies over the configuration space summarized in Table 8.

Modality We vary the input modality combinations available to agents: Text (T), Numerical (N), Visual (V), and Forecast (F). Combinations (e.g., TN, TVF) represent multimodal settings where agents can leverage multiple information sources.

Judge Tools We restrict the tools available to judge agents to isolate their impact: code execution only, numerical lookup only, or neither.

Agent Config We evaluate configurations with only judge agents (decision without debate) and without judges (pure agent interaction), to assess the necessity of hierarchical evaluation.

Components We remove specific components, including both tools, numerical lookup, or numerical judge lookup, to quantify their individual contributions.

A.5 Prompt Templates

All prompts are released with the code. We summarize the four core templates here. Modality-specific instantiations only differ in the `{modality_block}` slot, and the proposer, challenger, and analyst agents of a modality differ only in a short role instruction.

- **Agent (initial)**: "You are the {modality} agent for a multimodal time-series task. You are given the task question, output format, and the {modality}-specific evidence. Reason only from your modality. Return: (i) answer, (ii) confidence, (iii) modality-specific evidence."
- **Agent (refine, round n)**: "Below are the other agents' round $n-1$ outputs: {peer_outputs}. Your previous output was: {a_mˆ(n-1)}. Either defend, refine, or revise. Cite the specific evidence that changes (or fails to change) your position, and state whether your answer changed."
- **Judge (round k)**: "You receive: agent outputs {a_mˆ(N)} and peer judges' round $k-1$ views {g_{j'}ˆ(k-1)}. Evaluate each agent's evidence on quality (task-relevance, temporal consistency, evidential support), not popularity. Do not count votes; identify which modality evidence is most appropriate for the question type. Produce a structured judgment with selected answer, supporting evidence, rejected alternatives, and reasons."
- **Synthesizer**: "Given the judges' final-round outputs {g_jˆ(K)}, produce a single prediction in the required format. Prefer the answer with the strongest evidence, not necessarily the majority. If judges disagree, explain which judge output is best supported. Do not deliberate further."

The synthesizer is a single LLM call with no further deliberation. Representative trace excerpts in §G show it defaulting to majority-style outputs such as "$X/3$ say Y. Following majority.".

B LLM Disclosure Statement

This work studies LLM-based multi-agent systems as its primary research subject.

Use in Experiments All agents, judges, and synthesizers are instantiated using `gpt-4.1-mini` via OpenRouter, except for the Gemini 2.5 Flash replication in §F. Within each comparison, all methods use the same model to ensure fairness.

Use in Writing An LLM assistant was used solely for language editing and organization of the manuscript. All experimental results, analyses, and conclusions are derived from actual experiment logs and were not generated by the AI assistant.

C Ethical Considerations

This work aims to characterize the limitations of LLM-based multi-agent systems.

Potential Benefits and Risks By identifying failure modes and limitations, this work helps prevent misapplication of such systems and informs safer and more effective deployment in practice. We do not anticipate direct negative societal impacts, as the study focuses on documenting limitations rather than extending capabilities.

Data Considerations All experiments are conducted on publicly available benchmark datasets (MTBench, TimerBed, TSQA), none of which contain personally identifiable information (PII).

D Reproducibility Statement

We provide detailed information to facilitate reproducibility.

Model and Provider All experiments use `gpt-4.1-mini` accessed via OpenRouter, with temperature set to 0.0 for deterministic behavior. The cross-backbone analysis additionally uses `gemini-2.5-flash`, `gpt-5-mini`, and `grok-4.1-fast` via OpenRouter for the single-agent baseline. Its Gemini column uses the dated alias `google/gemini-2.5-flash-preview-09-2025`. The MAD-TS versus MM-SA replication uses the stable `google/gemini-2.5-flash` checkpoint for both methods.

Runs and Seeds All main results are averaged over three runs (run id = 1–3), corresponding to seeds 2026–2028, with 100 samples per task. HAR, EMG, and TEE use their full test splits of 96, 41, and 73 samples. Within a run, every method is evaluated on the same examples. Unparsable outputs score as incorrect, and no pairs are excluded. The ablation and sensitivity studies use 20 samples per task and report run 1. The 41 task-configuration pairs that also have a second or third run give the variability summarized in Table 9.

Compute Usage MAD-TS processes 1,910 samples across 20 tasks/run. Run 1 consumed 261.1M tokens and 29.2 hours of wall-clock time. Runs 2 and 3 consumed 335.3M and 333.5M tokens.

Datasets Experiments are conducted on MTBench, TimerBed, and TSQA benchmarks.

Code and Data The code, prompts, and analysis scripts are available at `https://github.com/itouchz/MAD-TS`.

Section	Median std	Max std	n
Main Table 2 – classification (3-run)	0.016	0.066	117
Main Table 2 – MCQA (3-run)	0.021	0.048	24
Main Table 2 – regression MAE (3-run)	9.99	182.87	56
Ablation/hyperparameter cls+MCQA (2–3 run)	0.050	0.125	41

Table 9: Standard deviation across runs by section. Ablation cells (20 samples/run) have $\sim 3\times$ the variability of the 3-run, 100-sample main-table baselines.

E Run-to-Run Variability

All methods in Table 2 are 3-run means with 100 samples per run (seeds 2026–2028). Table 10 lists MAD-TS by run. Its run-to-run std is at most 0.047 on the 13 classification/MCQA tasks. Table 12 gives the paired MAD-TS−MM-SA test for every task and run. On the regression tasks, the paired MAD-TS−MM-SA MAE differences by run are $+0.15$, $+1.47^{*}$, and $+1.96^{*}$ for Finance Forecasting, $+0.76^{*}$, $+0.82^{*}$, and $+0.45$ for Weather Forecasting, $+62.50^{*}$, $+6.37$, and -0.10 for TSQA Forecasting, and $+20.63$, $+4.85$, and $+29.24$ for TSQA Imputation. A star marks a 95% paired-bootstrap interval that excludes zero, and all five starred differences favor MM-SA. In run 1, the two rows where MAD-TS is uniquely best are not significant against the strongest competitor. On CTU the difference to MAD is $+0.01$ with CI $[-0.05, +0.07]$, and on TSQA Classification the difference to MM-MAD is $+0.02$ with CI $[-0.09, +0.13]$. The ablation and sensitivity results (Figures 2 and 3 and Table 4) report run 1 at 20 samples per task. Table 9 summarizes the standard deviation across runs. Baseline classification/MCQA accuracies are tight, below 0.07 in the worst case. Regression MAE swings are large, because parse failures on individual samples yield order-of-magnitude variation. Ablation and hyperparameter cells have a median std of 0.05 and a worst case of 0.125 across the 41 task-config pairs with two or more runs. This larger ablation variability is consistent with the framework's sensitivity to its configuration reported in §3. Many apparent differences across these configurations are within or close to the run-to-run noise.

F Cross-Backbone Analysis

We also evaluate the text-only SA baseline on three additional backbones, `gemini-2.5-flash`, `gpt-5-mini`, and `grok-4.1-fast`, on the 13 classification/MCQA tasks where the correctness of each sample is unambiguous (Table 11). No back-

Task	Run 1	Run 2	Run 3	Mean ± std
Finance Trend	0.420	0.390	0.470	0.427±0.033
Finance Correlation	0.300	0.280	0.300	0.293±0.009
Finance MCQA	0.800	0.800	0.810	0.803±0.005
Weather Trend	0.370	0.280	0.280	0.310±0.042
Weather MCQA	0.620	0.590	0.570	0.593±0.021
ECG	0.240	0.250	0.260	0.250±0.008
HAR	0.229	0.292	0.344	0.288±0.047
EMG	0.244	0.195	0.268	0.236±0.030
CTU	0.540	0.560	0.510	0.537±0.021
RCW	0.420	0.510	0.480	0.470±0.037
TEE	0.219	0.178	0.178	0.192±0.019
Anomaly Detection	0.490	0.570	0.570	0.543±0.038
Classification	0.630	0.660	0.630	0.640±0.014
Finance Forecasting (↓)	1.46	3.25	3.69	2.80±0.97
Weather Forecasting (↓)	4.23	4.16	4.24	4.21±0.03
Forecasting (↓)	224.64	55.21	53.87	111.24±80.19
Imputation (↓)	28.95	10.72	35.24	24.97±10.40
Finance Indicator (↓)	7.30	3.77	8.90	6.66±2.15
Weather Indicator, diff (↓)	4.62	5.68	5.49	5.26±0.46
Weather Indicator, max (↓)	4.22	4.81	5.44	4.82±0.50
Weather Indicator, min (↓)	3.66	3.31	3.60	3.52±0.15
QA, true/false	0.67	0.46	0.61	0.58±0.09
QA, multiple-choice	0.38	0.36	0.31	0.35±0.03

Table 10: MAD-TS results by run on all 20 tasks (accuracy ↑ unless marked MAE ↓, `gpt-4.1-mini`, seeds 2026–2028). QA rows cover the true/false and multiple-choice subtypes only.

bone uniformly dominates. `gemini-2.5-flash`, `gpt-5-mini`, and `gpt-4.1-mini` are each best on four of the 13 tasks, and `grok-4.1-fast` is best only on EMG. `gemini-2.5-flash` leads on Weather Trend at 0.56 against 0.25 for `gpt-4.1-mini`, but it trails on Finance MCQA at 0.70 against 0.82 and on TSQA Classification at 0.11 against 0.21. The spread across backbones on a single task reaches 38 pp on TSQA Classification and 31 pp on Weather Trend.

Table 13 tests the main comparison directly under a second backbone. We run MAD-TS and MM-SA with Gemini 2.5 Flash on the 1,210 examples of run 1, with the same code and analysis pipeline. Again, exactly one of 13 tasks shows a significant MAD-TS win, Finance MCQA, at +23 pp with Holm $p = .0002$. The `gpt-4.1-mini` win on TSQA Classification does not transfer. The micro-average difference is −2.6 pp and is not significant. Gemini MAD-TS abstains on 18.6% of samples, including 90 of 100 on TSQA Classification, while MM-SA answers every input. Abstentions score as incorrect. When we exclude them (n=985), the Finance MCQA win persists at +26 pp with $p < .0001$, only HAR remains a significant loss at −13 pp with $p = .036$, and the micro-average is +3.0 pp with CI $[-0.3, +6.4]$, which is not significant. Among final MAD-TS errors on the 9

Task	`gpt-4.1-mini`	`gemini-2.5-flash`	`gpt-5-mini`	`grok-4.1-fast`
Finance Trend	0.32 (±0.02)	**0.38**	0.33	0.29
Finance Correlation	0.23 (±0.02)	**0.27**	0.26	0.21
Finance MCQA	**0.82** (±0.02)	0.70	0.71	0.69
Weather Trend	0.25 (±0.01)	**0.56**	0.28	0.47
Weather MCQA	**0.62** (±0.05)	0.50	0.53	0.49
ECG	0.22 (±0.02)	**0.26**	0.22	0.23
HAR	0.25 (±0.00)	0.31	**0.33**	0.19
EMG	0.15 (±0.01)	0.15	0.17	**0.29**
CTU	0.51 (±0.02)	0.51	**0.58**	0.51
RCW	**0.47** (±0.03)	0.35	0.43	0.24
TEE	0.13 (±0.02)	0.15	**0.16**	0.14
TSQA Anomaly	**0.54** (±0.06)	0.50	0.44	0.50
TSQA Classification	0.21 (±0.03)	0.11	**0.49**	0.23

Table 11: Single-agent accuracy across four backbones on the 13 classification/MCQA tasks. The `gpt-4.1-mini` column is the mean and standard deviation over three runs and equals the SA column of Table 2. The other backbones have one run each. No backbone uniformly dominates, and the spread across backbones reaches 38 pp on a single task, so capacity-driven gains are highly task-dependent.

tasks of Table 5, all three judges give the same wrong answer in 96.3% of cases (543/564). The corresponding rate for `gpt-4.1-mini` in run 1 is 84.0% (409/487). Both backbones are homogeneous. These results therefore show that the shared-bias regime recurs under a second backbone, but they do not test heterogeneous-backbone agents, which remain the most direct follow-up (§4).

G Extended Failure Trace and Analysis

To better understand the limitations of our framework, we conducted a trace-based analysis using saved agent, judge, and synthesizer trajectories from our evaluation runs. We manually inspected a subset of representative failure cases to identify recurring reasoning patterns and systemic breakdowns. This qualitative analysis complements the quantitative results in the main paper by revealing *how* and *why* errors arise. The reading was exploratory. We inspected the 679 error traces of run 1 on the 13 classification/MCQA tasks without a fixed selection protocol or inter-rater coding. The taxonomy is therefore a descriptive lens rather than a prevalence estimate. The counts in §G.4 are the only quantitative result.

G.1 Representative Failure Cases

We begin with two illustrative examples that highlight distinct but recurring failure mechanisms. These cases demonstrate how correct reasoning can emerge locally (e.g., within a single judge) but still be overridden by flawed aggregation or interpretation strategies at the system level.

Task	Run 1		Run 2		Run 3	
	Diff. [95% CI]	p	Diff. [95% CI]	p	Diff. [95% CI]	p
Finance Trend	−0.040 [−0.14, +0.06]	0.828	+0.000 [−0.11, +0.11]	0.572	+0.010 [−0.10, +0.12]	0.500
Finance Correlation	−0.030 [−0.11, +0.05]	0.834	−0.060 [−0.14, +0.02]	0.952	+0.030 [−0.02, +0.08]	0.227
Finance MCQA	−0.030 [−0.10, +0.04]	0.867	−0.060 [−0.13, +0.01]	0.971	−0.040 [−0.12, +0.04]	0.881
Weather Trend	−0.130 [−0.25, −0.01]	0.986	−0.270 [−0.39, −0.15]	1.000	−0.240 [−0.35, −0.13]	1.000
Weather MCQA	−0.020 [−0.11, +0.07]	0.738	−0.100 [−0.20, +0.00]	0.982	−0.060 [−0.15, +0.03]	0.924
ECG	+0.000 [−0.07, +0.07]	0.613	−0.010 [−0.05, +0.03]	0.812	+0.010 [−0.04, +0.06]	0.500
HAR	−0.021 [−0.12, +0.08]	0.729	+0.073 [−0.02, +0.17]	0.105	+0.115 [+0.02, +0.21]	0.017
EMG	+0.024 [−0.15, +0.20]	0.500	−0.024 [−0.22, +0.17]	0.685	+0.000 [−0.20, +0.20]	0.598
CTU	+0.050 [−0.06, +0.16]	0.243	+0.060 [−0.05, +0.17]	0.189	−0.010 [−0.12, +0.10]	0.636
RCW	−0.040 [−0.11, +0.03]	0.927	−0.010 [−0.07, +0.05]	0.746	+0.010 [−0.05, +0.07]	0.500
TEE	+0.041 [−0.07, +0.15]	0.304	−0.027 [−0.14, +0.08]	0.760	+0.000 [−0.12, +0.12]	0.588
TSQA Anomaly	−0.090 [−0.19, +0.01]	0.974	−0.030 [−0.13, +0.07]	0.788	+0.000 [−0.11, +0.11]	0.572
TSQA Classification	+0.420 **[+0.30, +0.54]**	**0.000**	+0.390 **[+0.27, +0.51]**	**0.000**	+0.380 **[+0.25, +0.51]**	**0.000**

Table 12: Paired MAD-TS−MM-SA accuracy differences by run on the 13 classification/MCQA tasks with `gpt-4.1-mini`. The 95% CI uses 100,000 paired bootstrap resamples. The p value is the exact one-sided McNemar test of MAD-TS superiority without correction. Bold marks the only differences that remain significant after Holm correction within a run.

Stock Trend Prediction

Ground Truth: +2% to +4% | **Prediction:** >+4% (incorrect). This example illustrates a *majority-over-quality failure*: although one judge correctly synthesizes multimodal evidence, the final decision is dominated by majority voting.

```
TEXT:    ">+4%" (bullish sentiment, forward-looking)
VISUAL:  "<-4%" (downward chart pattern)
NUMERICAL: "-2% to +2%" (oscillations, no clear trend)

JUDGE_0: TEXT -> ">+4%"
JUDGE_1: SYNTHESIZE -> "+2% to +4%" [correct] (correctly
     balanced!)
JUDGE_2: TEXT -> ">+4%"

SYNTHESIZER: "2/3 say >+4%. Following majority." -> ">+4%"
      [incorrect]
```

Weather Trend Prediction

Ground Truth: stable | **Prediction:** decreasing (incorrect). This example demonstrates a *momentum fallacy*, where large historical changes dominate reasoning despite clear recent stabilization.

```
Data: Temperature 30 degC -> 15 degC (7 days), then
     stabilized 14-16 degC (3 days)

NUMERICAL:
  "Start=30 degC, End=15 degC. Difference=-15 degC. Trend:
       decreasing."
  <- Momentum Fallacy: ignores stabilization

VISUAL:
  "Chart shows decline then stabilization. Trend: stable."
       [correct]

JUDGE_0: "Numerical is precise. Decreasing."
JUDGE_1: "Visual shows recent plateau. Stable." [correct]
JUDGE_2: "-15 degC is significant. Decreasing."

SYNTHESIZER: "2/3 say decreasing." -> "decreasing" [
     incorrect]
```

These examples highlight a key theme: *correct reasoning signals are often present but systematically suppressed by aggregation mechanisms*.

G.2 Taxonomy of Failure Modes

From this exploratory reading, we identify 15 recurring failure modes, organized according to the level at which they arise: (i) agent-level reasoning, (ii) judge-level evaluation, and (iii) synthesizer-level aggregation. This taxonomy provides a structured lens for understanding where and how errors are introduced in the pipeline.

Agent-Level Failures These failures originate from how individual modality-specific agents interpret and reason over inputs. They often involve misinterpretation of temporal patterns, over-reliance on partial evidence, or inappropriate generalization.

Judge-Level Failures Judge failures arise during the evaluation and comparison of agent outputs. These errors reflect deficiencies in meta-reasoning, such as over-reliance on agreement signals or inconsistent application of evaluation criteria.

Synthesizer-Level Failures These failures occur at the final aggregation stage, where the system must consolidate competing judgments into a single prediction. Errors at this level often override correct intermediate reasoning.

G.3 Failed Interventions

During development we tried 12 targeted prompt-engineering interventions designed to enforce better reasoning, consistency, and aggregation. Despite their intuitive appeal, none yielded consistent improvements. Instead, they often introduced new failure modes. Table 17 summarizes these attempts

Task	n	MAD-TS	MM-SA	Diff.	95% CI	Holm p
Finance Trend	100	0.410	0.310	+0.100	[−0.030, +0.220]	0.9028
Finance Correlation	100	0.310	0.330	−0.020	[−0.100, +0.060]	1.0000
Finance MCQA	100	0.810	0.580	+**0.230**	[+0.130, +0.330]	**0.0002**
Weather Trend	100	0.210	0.420	−0.210	[−0.330, −0.090]	1.0000
Weather MCQA	100	0.480	0.470	+0.010	[−0.070, +0.090]	1.0000
ECG	100	0.400	0.370	+0.030	[−0.080, +0.140]	1.0000
HAR	96	0.188	0.323	−0.135	[−0.240, −0.031]	1.0000
EMG	41	0.146	0.366	−0.220	[−0.366, −0.073]	1.0000
CTU	100	0.510	0.560	−0.050	[−0.190, +0.090]	1.0000
RCW	100	0.450	0.370	+0.080	[+0.000, +0.160]	0.5775
TEE	73	0.301	0.301	+0.000	[−0.096, +0.096]	1.0000
TSQA Anomaly	100	0.490	0.530	−0.040	[−0.140, +0.060]	1.0000
TSQA Classification	100	0.030	0.250	−0.220	[−0.320, −0.130]	1.0000
Micro-average	1210	0.377	0.402	−0.026	[−0.056, +0.005]	–

Table 13: Gemini 2.5 Flash replication of MAD-TS versus MM-SA on the 13 classification/MCQA tasks with 1,210 paired examples (accuracy). Differences are computed before rounding. The Holm column tests MAD-TS superiority only with an exact one-sided McNemar test. Bold marks the only difference significant after Holm correction. MAD-TS abstentions score as incorrect.

Failure	Description
Evidence Abandonment	Agents concede when outnumbered, even with strong evidence
Momentum Fallacy	Extrapolating past trends as future predictions
Temporal Confusion	Conflating past observation with future prediction
False Precision	Precise calculations with inappropriate confidence
Speculation as Fact	Treating analyst opinions as verified data
Fixed Window Blindness	Claims about trends from limited data window
Anomaly Position Blindness	Detecting anomalies without positional context (peak vs trough)
Turning Point Blindness	Continuing past patterns despite regime changes

Table 14: Agent-Level Failures (8 types)

Failure	Description
Vote-Counting	Counting agreement instead of evaluating quality
Question Misclassification	Wrong categorization → wrong modality priority
Ignoring Own Classification	Correctly classifying, then violating in decision
Wrong Synthesis/Selection	Blending when should select, or vice versa
Confidence Asymmetry	Numerical sounds authoritative → inappropriate deference

Table 15: Judge-Level Failures (5 types).

Failure	Description
Majority Over Quality	Following 2/3 vote despite better minority reasoning
Ignoring Contradictions	Accepting answers contradicting stated reasoning

Table 16: Synthesizer-Level Failures (2 types).

Intervention	Intended Effect	Actual Effect
Question type classification	Better modality matching	Misclassification errors
Evidence integrity rules	Agents hold position	Quoted but not followed
Hard constraints in judge	Force consistency	Stated then violated
Momentum fallacy warnings	Stop extrapolation	Acknowledged, not applied
Confidence weighting	Quality over votes	All claimed high confidence
Task-specific rules	Accuracy by task	Fixed one, broke others
Turning point logic	Recognize regime changes	Ignored; calculations preferred
False precision warnings	Trust patterns over math	Acknowledged, not applied
Complementary synthesis	Explain modality roles	Quoted then vote-counted
Analyst causal logic	"Cause removed → effect stops"	Acknowledged, not applied
Modality profile enhancements	Better modality behavior	Mixed; judges still vote-counted
Synthesizer quality evaluation	Evaluate reasoning, not votes	All judges claimed high quality

Table 17: **The whack-a-mole pattern.** Every targeted fix moved errors around rather than eliminating them.

qualitatively. We did not retain accuracies for each intervention, so the table is an observation from development rather than a measured comparison. A common pattern emerges. Interventions are *acknowledged at the language level*, where models repeat the rule, but they are *not enforced at the decision level.*

G.4 Quantitative Failure Attribution

Table 18 reports the breakdown by task behind Table 5, pooled over three runs. Across nine classification/MCQA tasks with 2,688 traces and 1,447 errors, all three judges agreed on the same wrong answer in 85.0% of error cases. The rates by run are 84.0%, 83.3%, and 87.8%. This leaves the synthesizer no useful disagreement to arbitrate. Only 11.1% of errors involved judges disagreeing, and within these at most 5.3% had a correct judge overridden in synthesis. Unanimity exceeds 90% on Finance Correlation, Weather MCQA, TSQA Anomaly, and TSQA Classification. It is lowest on the two trend-prediction tasks, at 73–76%, where judges most often split, and on ECG, at 75%, where judge answers most often fail to parse. The three judges share the same backbone, prompt, and greedy decoding, so unanimity is repeated-output agreement rather than agreement among independent annotators. The quantitative attribution complements the qualitative taxonomy in §G.2. Vote-

Task	Traces	Errors	Unanimous wrong	All wrong, split	≥1 judge correct	Unparsable
Finance Trend	300	172	131 (76%)	19	22	0
Weather Trend	300	207	152 (73%)	30	23	2
Finance Correlation	300	212	193 (91%)	12	7	0
Finance MCQA	300	59	50 (85%)	0	9	0
Weather MCQA	300	122	115 (94%)	2	5	0
ECG	300	225	168 (75%)	6	3	48
HAR	288	205	182 (89%)	13	4	6
TSQA Anomaly	300	137	135 (99%)	0	2	0
TSQA Classification	300	108	104 (96%)	2	2	0
Total	**2,688**	**1,447**	**1,230 (85%)**	**84**	**77**	**56**

Table 18: Error attribution by task, pooled over three runs. The four right-hand columns partition the errors. Unparsable means that at least one judge's answer could not be parsed.

Factor	Classification	Regression
More modalities (Figure 2)	No monotonic gain, full TNVF never best	Best is a 1–2 modality subset
More judge rounds (Figure 3, left)	Task-dependent, direction reverses	Mostly worse on forecasting
More debate rounds (Figure 3, right)	Non-monotonic, task-dependent	Forecasting improves, indicator degrades

Table 19: Directional tendencies in the targeted ablations with 20 samples per task.

counting and majority-over-quality are present but rare. The dominant pattern is that the modality agents and judges share the same prior.

G.5 Root Cause: Unpredictable Task-Configuration Interactions

The core challenge is not any single failure mode, but the **unpredictable interaction between task characteristics and framework configurations** on top of a shared modality-agent bias. The same design choice (e.g., number of modalities or judge rounds) can produce opposite effects depending on the task. ***Why is this problematic?***

1. Practitioners cannot determine *a priori* which configuration will perform best.
2. Task-specific tuning undermines the value of using a general-purpose framework.
3. Improvements on one metric often come at the expense of another.
4. Shared upstream bias means that even "good" aggregation cannot recover correctness when no judge has the right answer.

More fundamentally, current LLM-based systems lack the *meta-cognitive capability* to adapt their reasoning and aggregation strategies dynamically based on task demands. As Table 5 shows, the agents and judges instantiated from the same backbone share the same blind spots, leaving the synthesizer with little leverage.

H Glossary

For clarity, Table 20 summarizes the key failure modes and reasoning patterns referenced throughout the paper.

Term	Definition
Evidence Abandonment	Agent concedes correct position when outnumbered
Momentum Fallacy	Extrapolating past trends as predictions
Temporal Confusion	Conflating past, present, and future
False Precision	Inappropriate confidence in precise calculations
Speculation as Fact	Treating opinions as verified data
Fixed Window Blindness	Ignoring limited data scope
Turning Point Blindness	Continuing patterns despite regime changes
Vote-Counting	Counting agreement over evaluating quality
Confidence Asymmetry	Precise language causing inappropriate deference
Cross-Modal Contamination	Debate causing modalities to converge
Whack-a-Mole Pattern	Fixes creating new failures elsewhere

Table 20: Glossary of reasoning and debate failure modes used in this work.