NEURAL INFORMATION PROCESSING SYSTEMS

Beyond LLM-Based Reasoning: Lightweight GNNs for Agent Failure Attribution

Ting-Wei Li, Yuanchen Bei, Xiao Lin, Hanghang Tong
University of Illinois Urbana-Champaign

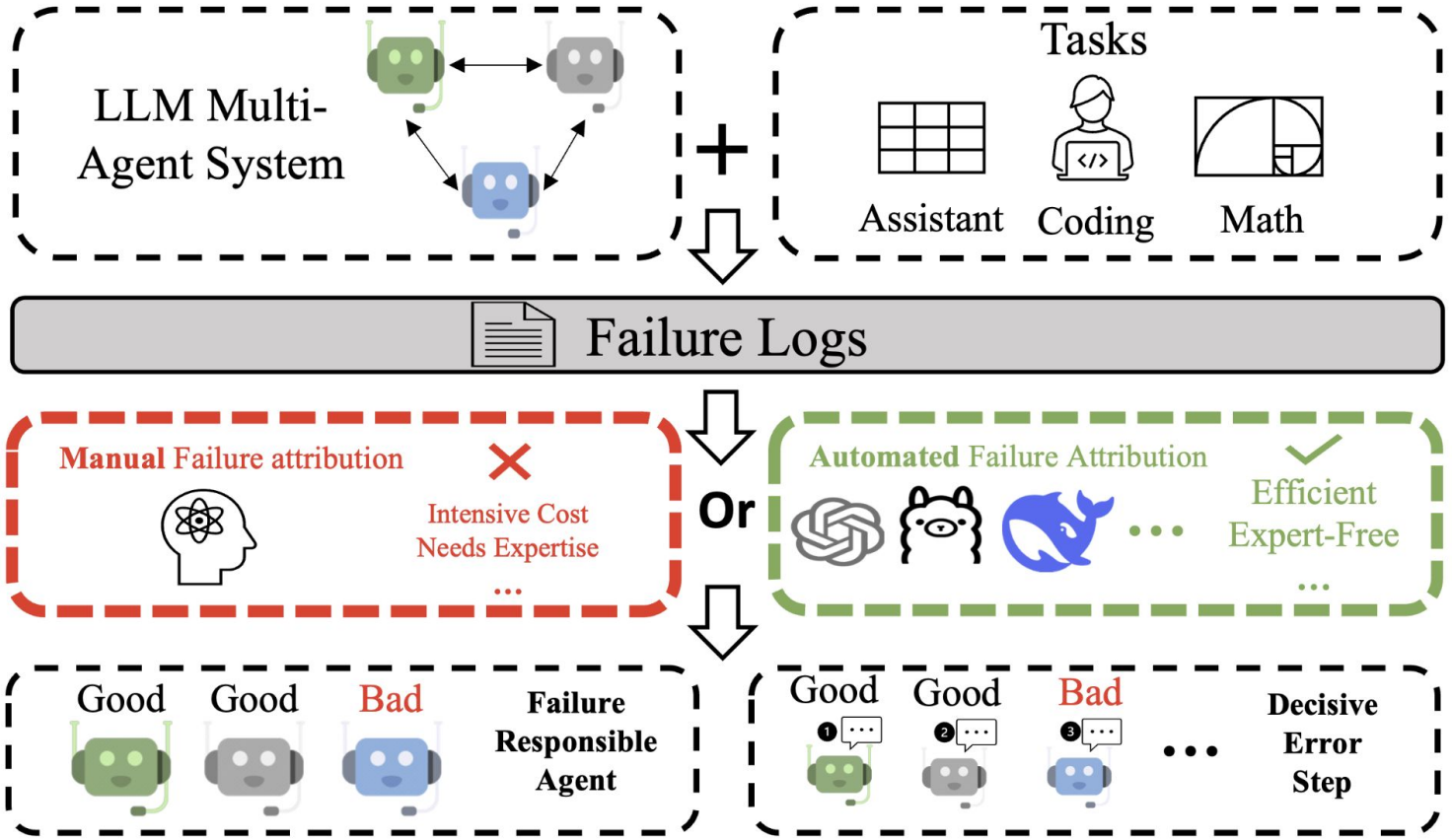
NeurIPS 2026 Poster



Outline

1. Motivation & Challenges
2. Problem Formulation & Pipeline
3. AFANet: A Lightweight GNN Architecture
4. Experimental Results
5. Conclusion & Future Directions

Motivation & Challenges



Why GNNs?

Graph-Native Architecture

Graph Neural Networks directly model agent dependencies and topology.

Fast & Accurate Attribution

Delivers lightweight, real-time diagnostic with minimal compute footprint.

Zhang, Shaokun, et al. "Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems." ICML 2026

Problem Formulation

1. MAS Trajectories

Interaction Execution Graph

Agent set A = {a_1, ..., aM} produces trajectory T = {(a_t, x_t)} over step horizon T for query q.

Task Outcome Verification

Verifier outcome o = g(T, q) ∈ {0, 1}, where o = 0 indicates overall system task failure.

2. Failure Attribution

Error Taxonomy & Labels

Predefined error types E = {e_1, ..., eK}. Label space defined as Cartesian product L = A × E.

Attribution Objective

Predict subset Y ⊆ L of (a_j, e_k) pairs identifying faulty agents and their specific error types.

3. Learning Task

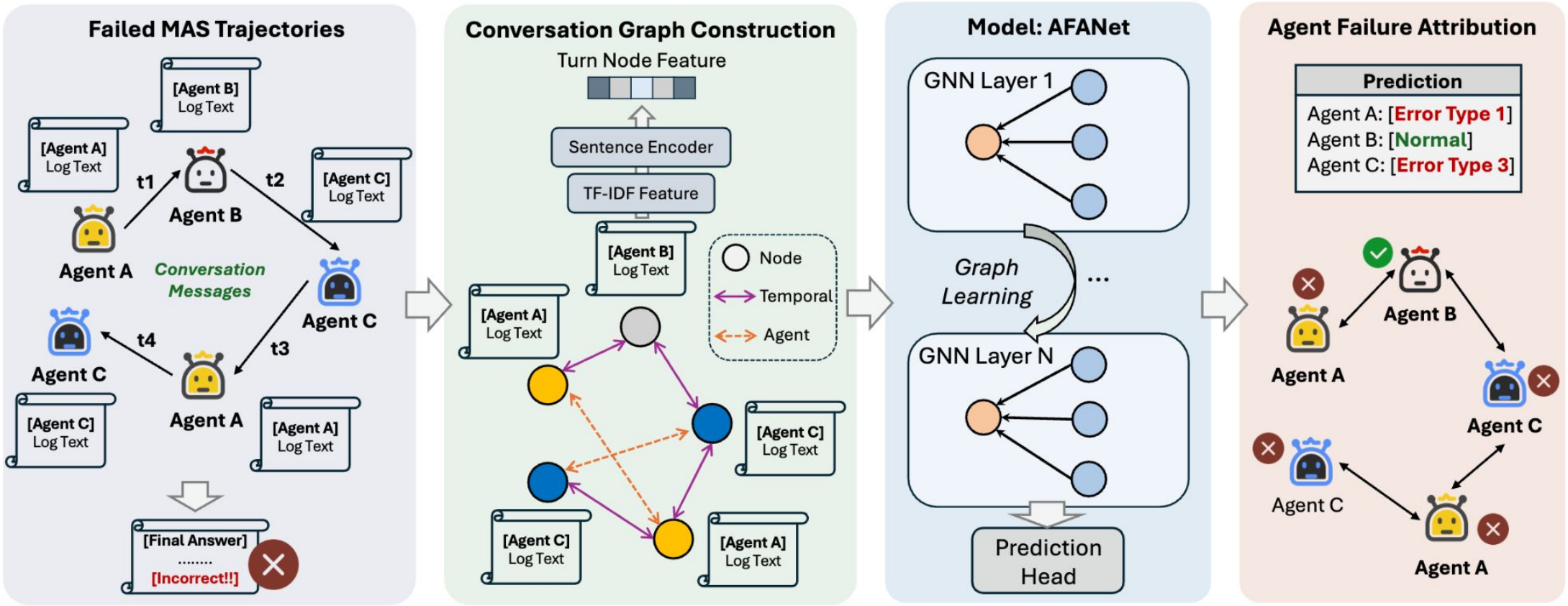
Multi-Way Classification

Learn mapping function h: T ⟶ {0, 1}^(M × (K+1)) across all agents in the trajectory.

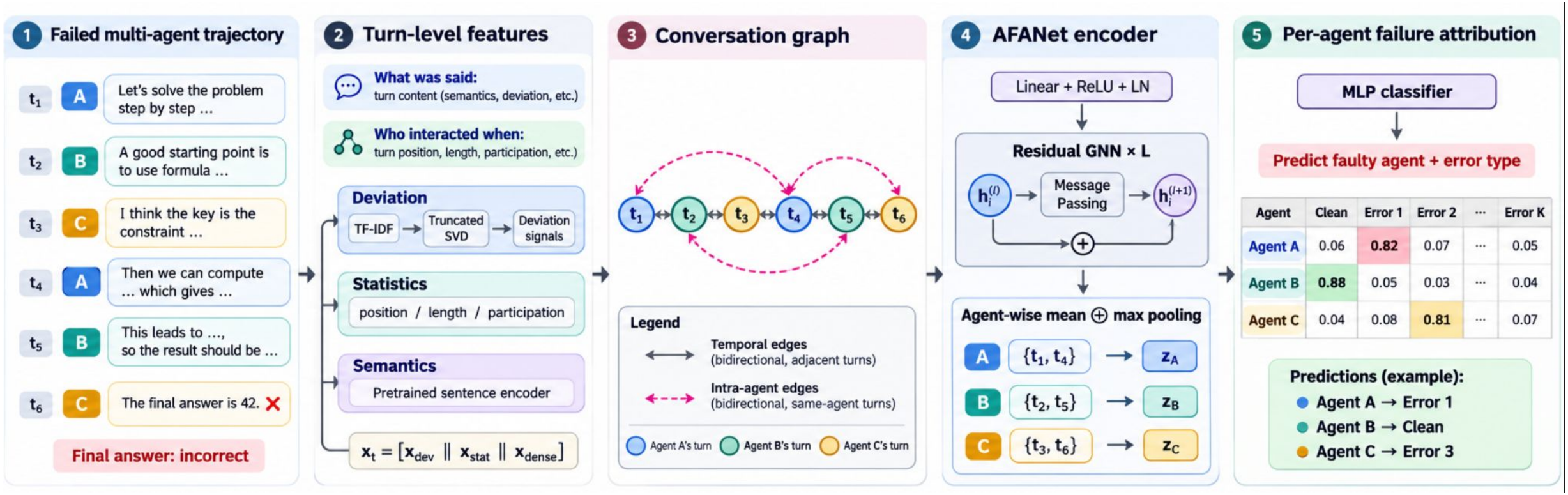
State Space Mapping

Class 0 represents clean/normal state; Classes 1...K correspond to specific error types in E.

Overall Pipeline



AFANet


1 Failed multi-agent trajectory
t1 A Let's solve the problem step by step ...
t2 B A good starting point is to use formula ...
t3 C I think the key is the constraint ...
t4 A Then we can compute ... which gives ...
t5 B This leads to ..., so the result should be ...
t6 C The final answer is 42.
Final answer: incorrect
2 Turn-level features
What was said: turn content (semantics, deviation, etc.)
Who interacted when: turn position, length, participation, etc.)
Deviation
TF-IDF
Truncated SVD
Deviation signals
Statistics
position / length / participation
Semantics
Pretrained sentence encoder
$x_t = [x_{dev} \| x_{stat} \| x_{dense}]$
3 Conversation graph
t1 t2 t3 t4 t5 t6
Legend
Temporal edges (bidirectional, adjacent turns)
Intra-agent edges (bidirectional, same-agent turns)
Agent A's turn
Agent B's turn
Agent C's turn
4 AFANet encoder
Linear + ReLU + LN
Residual GNN × L
$h_i^{(l)}$
Message Passing
$h_i^{(l+1)}$
Agent-wise mean ⊕ max pooling
A {t1, t4} → zA
B {t2, t5} → zB
C {t3, t6} → zC
5 Per-agent failure attribution
MLP classifier
Predict faulty agent + error type
Agent | Clean | Error 1 | Error 2 | ... | Error K
Agent A | 0.06 | 0.82 | 0.07 | ... | 0.05
Agent B | 0.88 | 0.05 | 0.03 | ... | 0.04
Agent C | 0.04 | 0.08 | 0.81 | ... | 0.07
Predictions (example):
Agent A → Error 1
Agent B → Clean
Agent C → Error 3

Experiment

Baselines:

- Direct Prompting
- Fine-tuned LLMs
- API models

AFANet matches or outperforms these baselines

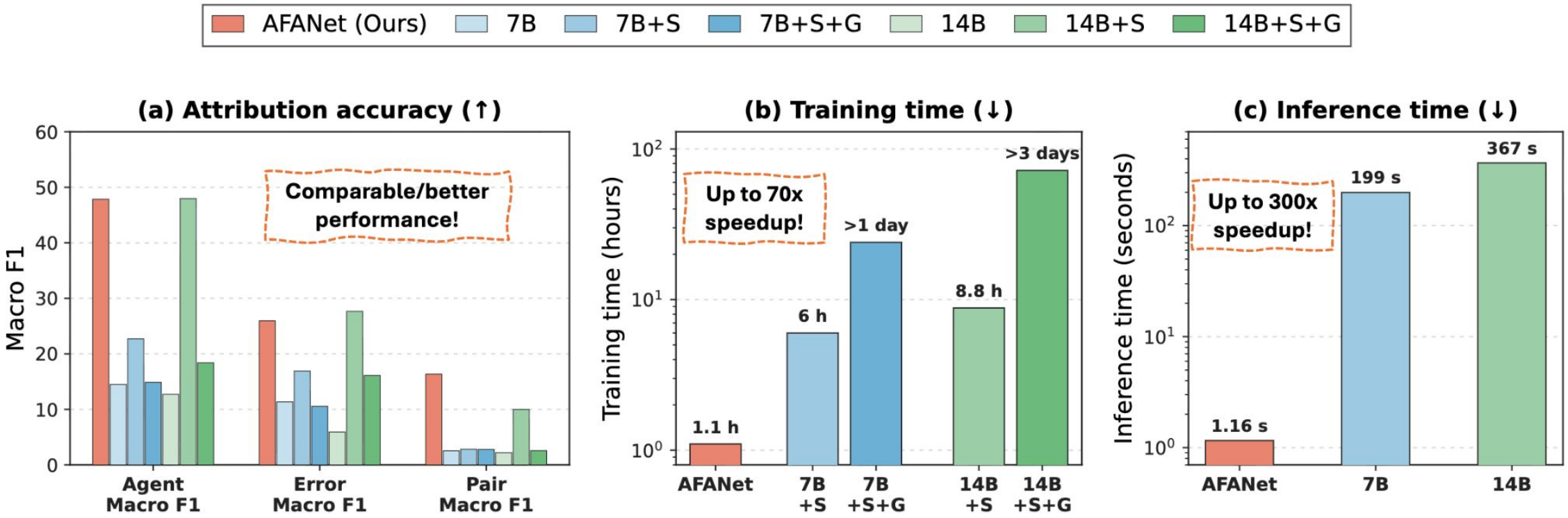
Category	Method	AEGIS-Bench						Who&When						Avg.
		Agent		Error		Pair		Agent		Error		Pair		
		μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	
Baseline	Random	4.54	3.56	11.23	11.15	0.33	0.21	1.06	0.83	8.74	7.14	0.11	0.05	4.08
Pre-trained LLMs	Qwen2.5-7B-It	27.55	14.49	14.96	11.36	5.02	2.52	40.92	23.50	3.64	1.77	2.31	1.14	12.43
	Qwen2.5-7B-It**	26.38	16.15	15.14	11.54	3.67	1.76	46.25	37.85	3.26	1.94	2.06	2.44	14.04
	Qwen2.5-14B-It	35.78	12.71	20.24	5.91	5.47	2.20	49.88	33.19	1.56	1.35	0.00	0.00	13.99
	Qwen2.5-14B-It**	36.72	19.62	18.59	16.49	5.86	2.61	45.47	44.83	0.00	0.00	0.00	0.00	15.85
	Qwen3-8B-NT	21.34	8.16	15.81	13.89	3.96	1.40	27.78	17.64	3.88	1.91	3.88	1.81	10.12
	Qwen3-8B-T	34.63	9.01	17.48	14.31	4.42	1.52	37.91	27.58	4.65	2.21	1.95	1.10	13.06
Fine-tuned LLMs	Qwen2.5-7B-It+S	60.03	22.70	19.61	16.90	5.05	2.80	43.51	32.51	6.77	4.20	1.26	0.52	17.99
	Qwen2.5-7B-It+S**	37.93	22.14	17.69	13.85	4.48	2.02	**56.77**	**65.81**	5.42	4.75	3.94	**4.58**	19.97
	Qwen2.5-7B-It+S+G	35.43	14.86	17.21	10.54	7.11	2.77	50.77	30.14	3.86	2.30	2.31	1.19	14.87
	Qwen2.5-14B-It+S	**76.53**	**47.97**	**27.53**	**27.66**	16.62	9.99	51.14	36.94	9.87	7.77	4.03	2.08	**26.51**
	Qwen2.5-14B-It+S**	41.04	21.80	17.16	13.83	4.38	2.02	49.37	64.90	1.77	1.73	0.41	0.25	18.22
	Qwen2.5-14B-It+S+G	49.74	18.38	21.19	16.10	6.84	2.55	54.43	40.88	4.15	2.67	2.45	1.49	18.41
	Qwen3-8B-NT+S	64.79	38.96	20.37	20.36	9.68	5.73	45.48	30.77	8.00	5.29	5.17	2.33	21.41
	Qwen3-8B-NT+S+G	45.91	17.39	20.89	15.15	6.94	2.82	50.94	38.26	2.21	1.68	2.21	1.45	17.15
	Qwen3-8B-T+G	36.11	15.73	17.94	12.03	4.41	1.66	53.12	40.52	11.25	6.91	**8.10**	3.19	17.58
Proprietary LLMs	GPT-4.1	37.48	11.12	20.65	15.75	7.44	2.27	42.29	28.93	7.00	5.84	3.36	1.16	15.27
	GPT-4o-mini	38.54	14.72	19.95	16.02	5.76	1.63	47.42	34.21	5.26	3.33	2.11	0.98	15.83
	o3	40.31	23.27	22.37	16.76	7.86	2.27	53.10	42.55	**14.88**	8.63	7.41	3.98	20.24
	Gemini-2.5-Flash	42.02	16.45	23.47	19.85	6.99	2.76	55.56	36.98	11.94	7.96	7.32	3.33	19.55
	Gemini-2.5-Pro	41.32	16.15	19.93	16.29	6.96	2.88	53.11	34.92	11.07	8.11	6.81	2.69	18.35
	Claude-Sonnet-4	40.73	15.51	21.21	16.55	7.68	2.34	44.76	37.23	13.33	**9.23**	6.77	2.66	18.16
Ours	`AFANet`	74.16	47.86	27.01	25.96	**17.42**	**16.35**	37.93	20.26	13.79	6.05	6.90	4.16	24.82

Experiment

Table 4: Ablation study of `AFANet`. Best and second-best results are in **bold** and underline. We show that graph structure, edge design and each feature component all contribute to `AFANet`'s performance. All scores are percentages (%). The reference model uses a 2-layer GCN with both edge types.

Setting	AEGIS-Bench						Who&When						Avg.
	Agent		Error		Pair		Agent		Error		Pair		
	μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	μF1	MF1	
Main Model													
GCN (all edges)	74.16	47.86	**27.01**	**25.96**	17.42	16.35	**37.93**	20.26	**13.79**	**6.05**	**6.90**	**4.16**	**24.82**
Edge Type Ablation													
GCN (no edges)	71.26	43.21	24.65	23.54	**18.49**	15.03	32.18	25.10	8.62	5.55	3.45	2.39	22.78
GCN (only same-agent edges)	70.53	42.74	22.32	21.58	16.31	12.23	36.78	26.21	6.32	2.95	4.60	3.16	22.14
GCN (only temporal edges)	**74.82**	**53.60**	24.82	24.39	17.54	**16.52**	35.06	20.41	5.75	3.41	4.60	1.99	23.58
Model Design Ablation													
W/o dev/stat features	65.90	35.47	23.59	22.25	14.00	9.80	**37.93**	**32.00**	5.75	2.89	1.72	1.14	21.04
W/o sentence embedding	73.58	48.18	23.62	22.14	14.78	10.16	**37.93**	21.76	8.05	5.44	2.30	1.33	22.44
W/o GNN	71.11	43.80	26.34	25.61	17.81	14.93	26.44	18.86	11.49	5.68	2.87	1.89	22.24

Experiment


AFANet (Ours)
7B
7B+S
7B+S+G
14B
14B+S
14B+S+G
(a) Attribution accuracy (↑)
Macro F1
Comparable/better performance!
Agent Macro F1
Error Macro F1
Pair Macro F1
(b) Training time (↓)
Training time (hours)
Up to 70x speedup!
1.1 h
6 h
>1 day
8.8 h
>3 days
AFANet
7B +S
7B +S+G
14B +S
14B +S+G
(c) Inference time (↓)
Inference time (seconds)
Up to 300x speedup!
1.16 s
199 s
367 s
AFANet
7B
14B

Conclusion

- MAS faulty detection does not necessarily need heavy LLM reasoning
- Lightweight graph model can predict well, with fraction of costs

Future directions:

- More MAS-related decision models
- More domain-robust algorithms

Thank you!

