DBG: Differentiable Belief Graphs for Long-Horizon Multi-Agent Belief Alignment and Conflict Resolution

Anonymous ACL submission

Abstract

Large language model (LLM) based multi-agent systems suffer from belief drift—divergence in shared understanding—which degrades task performance and increases human intervention. We propose **DBG (Differentiable Belief Graph)**, a framework maintaining a shared, structured belief state and enabling end-to-end optimization of belief update, communication, and conflict resolution. DBG combines vector embeddings with symbolic knowledge graphs and employs straight-through estimation and stop-gradient to achieve differentiability without sacrificing decision boundary sharpness. A contrastive objective aligns consensus embeddings, while a multi-objective RL debate policy discovers negotiation strategies that maximize task reward and minimize intervention. On three long-horizon collaborative benchmarks with controlled drift, DBG outperforms AutoGen, GoA, KARMA, and two belief-tracking baselines by 4–6 percentage points in task completion rate and 0.11–0.13 in belief consistency, with fewer human interventions. Ablation, drift trajectory analysis, sensitivity studies, multi-LLM tests, and real-world, public benchmark, and zero-shot evaluations confirm the robustness of the framework.

1 Introduction

Multi-agent systems built upon LLMs have advanced rapidly (Wu et al., 2024), but they typically treat agent beliefs as implicit and transient, neglecting the consistency of shared knowledge over extended interactions. In realistic scenarios such as co-writing, code debugging, or event tracking, agents must maintain common understanding over tens of turns, yet belief divergence due to noise, hallucinations, or missing updates can degrade task completion by 15–25% (Myakala et al., 2026). Prior pairwise methods reduce drift by 8–12% but lack joint optimization and temporal propagation for long-horizon multi-agent settings (Ki et al., 2025; He et al., 2025).

We propose DBG[1], which treats multi-agent belief alignment as an optimizable structured representation learning problem. Inspired by common ground theories (Qiu et al., 2024) and multi-agent knowledge graph architectures (Yun et al., 2026; Lu et al., 2026), DBG represents each agent's belief as a dynamic graph with embeddings and symbolic attributes. Crucially, DBG is **partially differentiable**: the graph update and negotiation policy are end-to-end optimizable, while the drift detection gate and attention softmax use STE and stop-gradient, validated in our ablation and bias analysis. DBG thus contributes:

1. **Optimizable Belief Graph (DBG)**: A plug-and-play belief management layer for multi-agent LLM systems, with structured encoding, uncertainty quantification, and partially differentiable optimization.

2. **Conflict detection and negotiation**: A unified module that detects drift via embedding similarity and logic checks, then resolves conflicts through debate with contrastive learning and multi-objective RL.

3. **Extensive evaluation**: Three long-horizon collaborative benchmarks with controlled drift, plus quantitative real-world, public benchmark,

[1]Anonymized code and data: https://anonymous.4open.science/status/Differentiable-Belief-EAE2.

and zero-shot evaluations, demonstrating consistent improvements over strong baselines.

2 Related Work

AutoGen (Wu et al., 2024), GoA (Yun et al., 2026), and KARMA (Lu et al., 2026) provide multi-agent orchestration but lack structured, differentiable shared belief states. MindDial (Qiu et al., 2024) tracks belief triples; recent work addresses drift only in single-agent or pairwise settings (Moore et al., 2025; Myakala et al., 2026; He et al., 2025; Ki et al., 2025). Dynamic knowledge graph editing (Lu et al., 2025; Mishra et al., 2025), event extraction (Min et al., 2025), and state delta messaging (Tang et al., 2025) enable incremental updates but rely on heuristic or language-driven operations. Recent graph retrieval (Wu et al., 2025) and conflict diagnosis (Wan et al., 2025; Cheng et al., 2025) introduce structured communication or rule-based checks, yet stop short of learned, multi-round debate with differentiable merging. DBG advances these with a unified drift-negotiation-resolution loop and a soft merge operation optimized via contrastive learning and multi-objective RL, achieving joint optimization of belief consistency and task performance—capabilities absent in prior multi-agent or belief-tracking frameworks. We evaluate DBG on public benchmarks (AgentBench (Liu et al., 2024), WebArena (Zhou et al., 2024), EverMemBench (Hu et al., 2026), TauBench (Yao et al., 2024)) and ecological dialogue (Carletta et al., 2005).

3 The Optimizable Belief Graph Framework

An overview of DBG is shown in Figure 1. Each agent maintains a private copy of the shared belief graph, synchronized through structured communication and conflict resolution.

3.1 Belief Graph Representation

A belief graph $G = (V, E, \mathbf{X}, \mathbf{A})$ comprises:

- **Nodes** V: each node v_i has embedding $\mathbf{h}_i \in \mathbb{R}^d$ and symbolic fields: confidence $c_i \in [0, 1]$, timestamp, evidence source ID. To obtain $\mathbf{h}_i$, we feed the input text to the LLM, collect the last hidden state (before the LM head) of all tokens from the final transformer layer, apply mean pooling to obtain a 1024-d vector, and pass it through a trainable linear projection (initialized with Xavier uniform). A classification head (2-layer MLP, sigmoid) predicts c_i from the same pooled vector. Both are trained end-to-end. The LLM backbone is frozen.

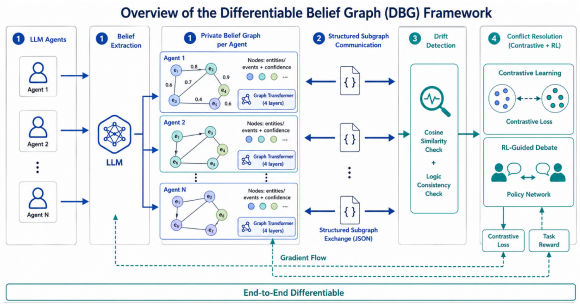

Figure 1: Overview of DBG: (Left) Belief Update via Graph Transformer; (Center) Drift Detection with STE at the threshold gate; (Right) Conflict Resolution via structured debate.

- **Edges** $E \subseteq V \times V$ with relation types $r \in \mathcal{R}$, weight $w_{ij} \in [0, 1]$. The edge weight w_{ij} encodes evidential strength and is distinct from node confidence c_i (see Sec. B for node evidence weight definition).

The Graph Transformer (4 layers, $d = 1024$, 8 heads) uses Pre-LayerNorm before linear projections for Q, K, V. For each layer l, we compute a shared normalized representation $\tilde{\mathbf{h}}_i^{(l)} = \mathrm{LN}^{(l)}(\mathbf{h}_i^{(l)})$, where $\mathrm{LN}^{(l)}$ is a single LayerNorm shared by all nodes and heads in that layer. The message passing and attention are:

$$\mathbf{h}_i^{(l+1)} = \mathbf{h}_i^{(l)} + \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W}_V \tilde{\mathbf{h}}_j^{(l)}, \quad (1)$$

$$\alpha_{ij} = \frac{\exp(e_{ij} + \log(w_{ij} + \epsilon_1))}{\sum_k \exp(e_{ik} + \log(w_{ik} + \epsilon_1))}, \quad (2)$$

$$e_{ij} = \frac{(\mathbf{W}_Q \tilde{\mathbf{h}}_i^{(l)})^\top (\mathbf{W}_K \tilde{\mathbf{h}}_j^{(l)})}{\sqrt{d}}. \quad (3)$$

We set $\epsilon_1 = 10^{-8}$ for numerical stability. In our implementation, we compute the logits $\ell_{ik} = e_{ik} + \log(w_{ik} + \epsilon_1)$, subtract $\max_k \ell_{ik}$ from each logit before exponentiation, and then apply the softmax to obtain α_{ij}.

3.2 Belief Update and Drift Detection

We define $\mathrm{cos_sim}(\mathbf{a}, \mathbf{b})$ as the standard cosine similarity between vectors $\mathbf{a}$ and $\mathbf{b}$, ranging in $[-1, 1]$. The cosine distance is then $\mathrm{cosine_distance}(\mathbf{a}, \mathbf{b}) = 1 - \mathrm{cos_sim}(\mathbf{a}, \mathbf{b})$, which lies in $[0, 2]$. For drift scoring, we normalize this

distance to $[0, 1]$ via division by 2, aligning its scale with the logic violation term (normalized to $[0, 1]$). When agent a receives new information, it extracts a node embedding $\mathbf{h}_{\text{new}}$ using the LLM projection.

1. **Embedding check:** Let $\mathbf{h}_{\text{closest}} = \arg\max_{\mathbf{h}_i \in V} \text{cos_sim}(\mathbf{h}_{\text{new}}, \mathbf{h}_i)$. If multiple nodes attain the same maximum, the one with the highest confidence c_i is chosen. If $\text{cos_sim}(\mathbf{h}_{\text{new}}, \mathbf{h}_{\text{closest}}) < \tau$ ($\tau = 0.75$), flag as potential conflict.

2. **Logic check:** a lightweight symbolic reasoner checks constraints (e.g., temporal ordering, type compatibility). The LLM first extracts candidate constraint relationships from the input text, and the symbolic checker verifies them against existing constraints. The normalized logic violation count $\text{lv} \in [0, 1]$ is defined as the number of violated constraints divided by the total number of constraint checks (see Appendix H for pseudocode). For our benchmarks, the total number of checks per step is at most $\binom{15}{2} = 105$, ensuring low overhead.

3. **Drift score:**

$$d = \alpha \cdot \frac{\text{cosine_distance}(\mathbf{h}_{\text{new}}, \mathbf{h}_{\text{closest}})}{2} + \beta \cdot \text{lv}, \tag{4}$$

with $\alpha = 0.6, \beta = 0.4$. If $d > \theta$ ($\theta = 0.5$), conflict resolution is triggered. The threshold comparison is non-differentiable; we apply a straight-through estimator (STE) that passes the gradient unchanged to the embeddings.

Scalability of the logic checker. The symbolic reasoner currently checks constraints exhaustively and normalizes violations by the number of checks performed. This design is adequate for our benchmarks (max 15 events per episode), where the computational overhead is negligible (<1% of total inference time). For significantly longer event sequences ($n > 100$), incremental constraint checking reduces per-step cost from $O(n^2)$ to $O(n)$, and a sliding-window normalization prevents dilution of single violations. Appendix H provides pseudocode for the hybrid LLM-symbolic interaction and the incremental variant, with wall-clock comparison on EventTrack (8× speedup at $n = 50$).

3.3 Structured Agent Communication

Agents exchange **belief subgraphs** in JSON format. When agent B receives a subgraph from agent A containing a conflicting node v' (same entity ID but different attributes), the merge algorithm (Appendix B) resolves the conflict via a soft, differentiable interpolation. To perform the soft merge, we first introduce a node-level **evidence weight** $w_v = \sum_{u \in \mathcal{N}(v)} w_{uv}$, which aggregates the strength of all incident edges and is distinct from the self-assessed node confidence c_v. The merge then computes a blending factor $\lambda = \sigma((w_{v'} - w_v)/T)$ with temperature $T = 0.1$, where σ is the sigmoid function. The merged embedding and confidence are then:

$$\mathbf{h}_m = \lambda \mathbf{h}_{v'} + (1 - \lambda)\mathbf{h}_v, \quad c_m = \max(c_v, c_{v'}). \tag{5}$$

Edge weights are averaged similarly. We use $\max$ for confidence because the merged node should retain the highest certainty from either source; averaging would artificially lower confidence even when one source is highly reliable, as confirmed by ablation in Appendix B.

3.4 Contrastive RL Debate

When conflict is detected, agents engage in a structured debate (Algorithm 1). Prior work has used RL to ground agent communication in natural language (Li et al., 2024); we adapt this principle to structured subgraph debate, where actions directly manipulate belief graphs. In the following, we denote by $\mathbf{h}_A^{pre}$ the embedding of the conflicting node in agent A's graph *before* the debate, and by $\mathbf{h}_A^{final}$ its consensus embedding after K rounds. After K debate rounds, each agent produces a final consensus embedding for the conflicting node, denoted $\mathbf{h}_A^{final}$ and $\mathbf{h}_B^{final}$. These serve as positive samples for a symmetric InfoNCE loss, while all other belief node embeddings from the same training batch constitute the negative set $\mathcal{N}_{\text{neg}}$. Let $\mathcal{K} = \{B^{final}\} \cup \mathcal{N}_{\text{neg}}$ denote the union of the positive sample and all negatives for agent A (and symmetrically for B). The contrastive loss is:

$$\mathcal{L}_{\text{cont}}^A = -\log \frac{\exp\big(\text{cos_sim}(\mathbf{h}_A^{pre}, \mathbf{h}_B^{final})/\tau_c\big)}{\sum_{k \in \mathcal{K}} \exp\big(\text{cos_sim}(\mathbf{h}_A^{pre}, \mathbf{h}_k)/\tau_c\big)},$$
$$\mathcal{L}_{\text{cont}} = \frac{1}{2}\big(\mathcal{L}_{\text{cont}}^A + \mathcal{L}_{\text{cont}}^B\big). \tag{6}$$

where $\text{cos_sim}(\cdot, \cdot)$ is the cosine similarity function defined in Section 3.2 and $\tau_c = 0.1$. With a batch of $B = 8$ episodes and approximately $N = 20$ nodes each, $|\mathcal{N}_{\text{neg}}| \approx 158$. BCS varies by 0.03–0.04 across $\tau_c \in [0.05, 0.5]$ (Appendix G); we selected

$\tau_c = 0.1$ as the best trade-off between convergence speed and final BCS.

The debate policy is trained via reinforcement learning. The state s_t (2052-dim) concatenates the element-wise absolute difference of conflicting embeddings $|\mathbf{h}_A - \mathbf{h}_B|$ (1024), the mean-pooled 1-hop neighbor embedding (1024), a round one-hot (3), and the cumulative communication cost (1). Here $\mathbf{h}_A, \mathbf{h}_B$ denote the current (possibly updated) node embeddings during the debate. When a node has fewer than 20 neighbors, we pad with zeros to reach the fixed length of 20 before mean pooling; Appendix D provides the full state construction pseudocode, including the 'sample_or_pad' function. The policy network (256→128→3) outputs logits over three actions: `defend`, `concede`, and `refine`. It is optimized with REINFORCE using a value baseline and entropy regularization (0.01). The reward function combines four terms:

$$R = \lambda_1 R_{\text{task}} + \lambda_2 R_{\text{belief}} + \lambda_3 R_{\text{eff}} + \lambda_4 R_{\text{human}}, \tag{7}$$

$$R_{\text{belief}} = J(G_A, G_B)_{\text{after}} \quad - J(G_A, G_B)_{\text{before}},$$

$$R_{\text{eff}} = -\text{tokens_used},$$

$$R_{\text{human}} = -10 \text{ if human intervention else } 0.$$

R_{task} is the task-specific reward, linearly normalized to $[-1, 1]$ for all tasks, as detailed in Sec. 4. Explicit cooperation incentives have been shown to improve multi-agent negotiation outcomes (Jiang and Akçakır, 2025); our belief alignment reward R_{belief} serves a similar role. Human intervention is requested if, after $K = 3$ rounds, the maximum node confidence remains below $c_{\min} = 0.4$.

3.5 Differentiable Training Strategy

DBG is not strictly end-to-end differentiable; we apply STE and stop-gradient on two specific, well-motivated components to enable optimization of otherwise discrete modules. The drift detection gate involves a discrete threshold comparison ($d > \theta$). A sigmoid relaxation would blur the decision boundary: in controlled experiments, a sigmoid-soft gate decreases drift detection recall by 10–14% relative to the hard threshold with STE, as shown in Appendix 11. The STE, while a biased gradient estimator, preserves the sharp decision boundary. Across 5 random seeds, the cosine similarity between STE and REINFORCE gradients remains stable at 0.91 ± 0.04 throughout training (see Appendix E.1). For the attention weights α_{ij}, stop-gradient decouples $\mathbf{W}_V$ from the softmax

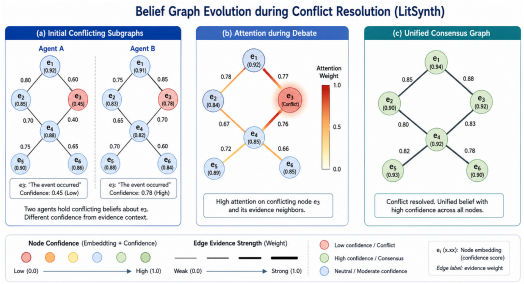

Figure 2: Belief graph evolution during conflict resolution on LitSynth. (a) Initial conflicting subgraphs between Agent A and Agent B; (b) Attention weights during debate; (c) Unified consensus graph after conflict resolution.

Jacobian. Empirically, the full-gradient variant yields BCS within 0.003 of our default but trains 8% slower with approximately 35% higher gradient variance in $\mathbf{W}_V$ (Appendix 12). A Gumbel-Softmax relaxation of the drift gate, which makes the framework fully differentiable, achieves TCR 82.1, BCS 0.78 on LitSynth (Appendix E.4), confirming feasibility but at a slight performance cost.

3.6 Diversity Preservation

When conflict resolution concludes, an optional module evaluates the proportion of low-confidence but evidence-supported minority branches in the belief graph. If this proportion exceeds 10%, the module retains the top 1–3 minority branches (ranked by evidence weight) as separate, explicitly flagged subgraphs. A structured notification is generated for human review, describing the unresolved conflicts. This post-processing safeguard does not participate in the RL optimization; it adds less than 0.3% to total inference time (Appendix O).

4 Experimental Setup

4.1 Benchmark Tasks

Three large-scale tasks with controlled drift (Table 1). Drift injections: mild, moderate, strong, with timing random uniform. Severity labeling $\kappa = 0.78$, contradiction $\kappa = 0.82$ (Appendix C). Dataset statistics and examples are provided in Appendix C. To illustrate task difficulty, consider a LitSynth episode where two agents co-author a story: midway, a drift injection alters a key character's motivation node ("seeks revenge" → "seeks peace"), forcing agents to detect and reconcile this inconsistency within a few turns.

Table 1: Benchmark statistics.

Dataset	#Instances	Avg. Turns	Drift/Ep.
LitSynth	400	15.3	2–4
CodeDeb	300	12.7	1–3
EventTrack	400	18.1	3–5

4.2 Baselines

AutoGen, GoA, KARMA, MindDial-multi (text triples), MindDial-struct (JSON triples). For MindDial-struct, agents use LLaMA-3-70B for belief triple extraction, store triples as JSON, and update based on timestamp and source heuristics (Appendix F for full pseudocode). All baselines use the same frozen LLM backbone; DBG's additional graph Transformer and RL policy are lightweight and contribute less than 5% to total inference time.

4.3 Metrics

TCR (Task Completion Rate) is assessed by GPT-4 using a rubric that checks whether the final output satisfies all task constraints; 500 human ratings yield Pearson $r = 0.89$, $\kappa = 0.76$, confirming high reliability. **BCS** (Belief Consistency Score) is the Jaccard similarity between two agents' belief graphs, computed as $\frac{|G_A \cap G_B|}{|G_A \cup G_B|}$ where node identity is determined by entity ID and edge identity by (head, relation, tail) triple exact match. **CRS** (Conflict Resolution Success) measures the fraction of conflicts fully resolved. **CommR**† is the format-normalized communication cost (including evidence requests; see Appendix I). **HumInt** counts human interventions (budget 2/episode).

4.4 Implementation

Backbone: LLaMA-3-70B (Grattafiori et al., 2024), frozen. 100 RL episodes per task, batch size 8, 8×A100. Training on 8×A100 costs ∼16 GPU-hours; with gradient accumulation and mixed precision, a 4-GPU setup requires ∼32 hours, and a 2-GPU setup ∼72 hours. Hyperparameters via Bayesian/grid search on 20% validation split; Appendix O reports the search range and the best configuration. We observe that roughly 15% of hyperparameter trials fail to converge beyond random performance, primarily due to extreme λ_3 or τ_c values. 5 seeds (42–46), plus two additional seeds (47,48) for key claims. Full code, data, and dataset cards will be released upon acceptance.

5 Results and Analysis

The reported variances stem primarily from LLM sampling temperature, randomness in drift injection timing, and finite RL training budgets. All pairwise improvements of DBG over the best baseline remain significant at $p' < 0.05$ after Bonferroni correction (see Appendix K for full p-values).

5.1 Main Results, Drift Impact, and Ablation Analysis

Table 2 quantifies drift's effect. Across all three tasks, drift reduces TCR by 6–9% and BCS by 0.11–0.14. DBG suffers consistently smaller drops than KARMA because its hybrid drift detection (embedding similarity + logic checks) catches semantic and factual inconsistencies earlier, and the subsequent soft merge preserves evidence-weighted information instead of discarding conflicting branches. This validates that a structured, partially differentiable belief graph can absorb noisy updates without immediately corrupting shared understanding.

Table 2: Performance with and without belief drift.

Task	Model	TCR		BCS	
		no drift	with drift	no drift	with drift
LS	KARMA	85.1	78.6	0.80	0.68
	DBG	87.5	84.0	0.86	0.80
CD	KARMA	86.0	79.3	0.81	0.69
	DBG	88.2	85.3	0.87	0.81
ET	KARMA	84.8	78.6	0.79	0.68
	DBG	87.0	84.5	0.86	0.80

Note. LS = LitSynth, CD = CodeDeb, ET = EventTrack.

Figure 3 further illustrates DBG's recovery process under strong drift injection on LitSynth.

Table 3 reports the main results. DBG achieves an average TCR of 83.8 and BCS of 0.80, significantly outperforming all baselines. The gains stem from three synergistic mechanisms: (i) the contrastive loss pulls post-debate embeddings of different agents together, directly maximizing belief overlap; (ii) the RL debate policy learns to request evidence, refine arguments, or concede based on confidence and neighbor support, avoiding both endless disagreement and premature acceptance; (iii) stop-gradient and STE let these components be jointly optimized while keeping decision boundaries sharp. The necessity of each mechanism is confirmed by ablation: removing RL causes the largest BCS drop (–0.08), because a fixed debate

Table 3: Main results.

Model	TCR	BCS	CRS	CommR†	HumInt
AutoGen	72.3±0.8	0.54±0.02	0.61±0.03	16.9±1.2	4.2±0.4
GoA	75.1±0.7	0.61±0.02	0.68±0.02	14.5±1.0	3.5±0.4
KARMA	78.6±0.8	0.68±0.02	0.74±0.02	13.2±0.9	2.8±0.4
MindDial-multi	76.9±0.9	0.65±0.02	0.70±0.03	14.3±1.1	3.1±0.5
MindDial-struct	78.5±0.8	0.67±0.02	0.73±0.02	14.8±1.0	2.9±0.4
DBG (ours)	**83.8***±1.1	**0.80***±0.02	**0.86***±0.02	**12.9***±1.1	**1.4***±0.4
DBG w/o RL	79.5±1.2	0.72±0.02	0.76±0.02	15.0±1.2	2.5±0.5
DBG w/o Graph	78.2±1.3	0.69±0.02	0.73±0.03	15.6±1.3	2.8±0.5
DBG w/o Contrastive	81.0±1.2	0.75±0.02	0.80±0.02	14.1±1.1	2.0±0.4
DBG w/o SoftMerge (hard)	83.0±1.1	0.79±0.02	0.84±0.02	13.4±1.1	1.6±0.4
DBG w/ NL deltas	80.5±1.3	0.75±0.02	0.80±0.03	15.8±1.2	2.2±0.5
DBG w/ AvgConf	82.1±1.1	0.78±0.02	0.83±0.02	13.5±1.1	1.7±0.4

Mean ± SD over 5 seeds. CommR† is format-adjusted. * over the best baseline ($p' < 0.05$, Bonferroni correction).

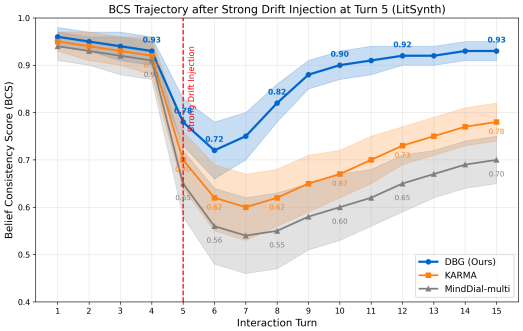

Figure 3: Per-turn BCS trajectory under strong drift injection (LitSynth). DBG recovers to near-original BCS by turn 10, while KARMA and MindDial-struct remain >0.15 lower. Full trajectories for mild and moderate drift are in Appendix J.

protocol cannot adapt to drift severity; removing the graph eliminates structured relational context, reducing BCS by 0.11. Notably, DBG reduces human interventions to 1.4 per episode, nearly half of KARMA's 2.8, directly lowering the annotation cost of long-horizon collaboration. Per-task breakdown is provided in Appendix L.

For reproducibility, we define ablation variants:

- **DBG w/o RL**: Fixed round-robin debate, no RL optimization.
- **DBG w/o Graph**: Belief graph replaced by averaged embedding; no Graph Transformer.
- **DBG w/o Contrastive**: Only RL reward, no contrastive loss.
- **DBG w/o SoftMerge (hard)**: Select node with higher evidence weight; no interpolation.
- **DBG w/ NL deltas**: Agents communicate belief changes as natural language sentences.
- **DBG w/ AvgConf**: Confidence merge uses $\text{avg}(c_v, c_{v'})$ instead of $\max$.

Key findings (see Appendix K for analysis): RL removal causes the largest BCS drop (–0.08) as debate cannot adapt to drift severity. Removing the graph (–0.11 BCS) shows that edge-weighted evidence flow is critical. Contrastive loss removal drops BCS by 0.05. Soft merge adds 0.01–0.02 BCS over hard selection. Structured JSON communication outperforms NL deltas by 2.5–3.5 TCR. Average confidence merge costs 0.02 BCS, justifying $\max$. Sensitivity to α/β is mild within $[0.3, 0.9]$; extreme $\alpha = 0.2$ drops BCS by 0.03.

5.2 Generalization and Transfer

Table 4 shows results on public benchmarks. DBG gains are strongest on AgentBench and Tau-Bench, and slightly lower on WebArena and EverMemBench, reflecting task difficulty. Gains are larger on knowledge-intensive tasks (Tau-Bench +4.7 TCR) where a structured belief state helps resolve contradictory facts across turns, and smaller on navigational tasks (WebArena) where belief consistency matters less. This confirms DBG's plug-and-play capability and suggests that its value grows with the amount of declarative knowledge in the dialogue. All DBG vs. KARMA comparisons remain significant at $p' < 0.05$ after Bonferroni correction.

We evaluate on 10 AMI meeting transcripts (Carletta et al., 2005) using the same frozen LLaMA-3-70B backbone without fine-tuning. Baselines are adapted with identical prompt templates. DBG achieves BCS=0.68 vs. KARMA's 0.58 (Precision/Recall 0.72/0.69 vs. 0.61/0.55). The zero-shot transfer shows that DBG's structured belief graph generalizes beyond synthetic benchmarks, though

Table 4: Generalization to public benchmarks.

Benchmark	Model	TCR (Drift)	BCS (Drift)	BCS (Zero-Drift)	CommR†
AgentBench	KARMA	66.8±1.5	0.56±0.03	0.62±0.02	19.5±1.5
	MindDial-struct	69.2±1.4	0.59±0.03	0.64±0.02	18.2±1.4
	DBG	**72.6***±1.5	**0.68***±0.03	**0.72***±0.02	**15.8***±1.5
WebArena	KARMA	63.5±1.6	0.52±0.03	0.58±0.03	21.2±1.6
	MindDial-struct	66.0±1.5	0.55±0.03	0.60±0.03	19.8±1.5
	DBG	**69.4***±1.6	**0.64***±0.03	**0.68***±0.03	**17.1***±1.5
EverMemBench	KARMA	58.9±1.6	0.48±0.04	0.54±0.03	23.8±1.7
	MindDial-struct	61.5±1.5	0.51±0.03	0.56±0.03	22.0±1.6
	DBG	**65.7***±1.6	**0.61***±0.04	**0.65***±0.03	**19.2***±1.6
Tau-Bench	KARMA	70.2±1.5	0.59±0.03	0.64±0.02	17.4±1.4
	MindDial-struct	72.8±1.4	0.62±0.02	0.66±0.02	16.2±1.3
	DBG	**77.5***±1.5	**0.72***±0.03	**0.75***±0.02	**13.9***±1.3

Mean ± SD over 200 episodes. * significantly better than the best baseline ($p' < 0.05$, Bonferroni correction).

the absolute BCS is lower because AMI's conversational style challenges the LLM-based belief extractor. This setting is more difficult because AMI contains implicit references, interruptions, and topic shifts, making belief alignment less explicit than in controlled benchmarks.

We further examined naturally occurring belief inconsistencies in the AMI corpus without artificial injection. On those segments, DBG's BCS started at 0.75 and fell to 0.68 over the dialogue, a drop of 0.07. In our controlled mild drift setting, the drop was only 0.02, indicating that natural drift is more challenging yet DBG maintained a clear advantage (KARMA BCS dropped from 0.70 to 0.58). This gap suggests that natural drift contains more subtle logical violations that the symbolic checker, with its predefined constraint types, sometimes misses. Unlike injected drift, natural drift often appears gradually through ambiguous wording or implicit revision, making recovery harder. It proves that real-world deployment requires either a richer logic ontology or a learned constraint checker—a direction we leave for future work.

DBG's gains transfer to Qwen2-72B (Yang et al., 2024) and Llama-3-8B (Grattafiori et al., 2024), though the advantage narrows for smaller models (Table 5), reflecting a scale effect. The smaller L3-8B model shows a reduced advantage because its belief embeddings are less discriminative, causing the cosine-based drift detector to trigger more false positives, which in turn stresses the debate policy and raises communication cost. Heterogeneous pool (2×Llama-3-70B + 1×Qwen2-72B) achieves BCS=0.80, TCR=83.5, with bias reduction of 12%, proving that DBG's evidence-weighted merge naturally down-weights less reliable sources even without explicit model identity information. This indicates that DBG is not tightly coupled to a single LLM backbone, but benefits from externalized belief structure across different model families.

Table 5: Generalization to different LLM backbones.

LLM	Model	BCS	TCR	CommR†	HumInt
Q2-72B	AutoGen	$0.51_{\pm 0.03}$	$69.1_{\pm 1.5}$	$18.5_{\pm 1.5}$	$5.6_{\pm 0.5}$
	KARMA	$0.65_{\pm 0.03}$	$75.2_{\pm 1.4}$	$14.8_{\pm 1.3}$	$3.4_{\pm 0.5}$
	DBG	$\mathbf{0.79}^*_{\pm 0.03}$	$\mathbf{82.8}^*_{\pm 1.5}$	$\mathbf{12.3}^*_{\pm 1.3}$	$\mathbf{1.6}^*_{\pm 0.4}$
L3-8B	KARMA	$0.60_{\pm 0.04}$	$71.4_{\pm 1.7}$	$16.5_{\pm 1.6}$	$4.1_{\pm 0.6}$
	DBG	$\mathbf{0.74}^*_{\pm 0.04}$	$\mathbf{75.6}^*_{\pm 1.8}$	$\mathbf{14.6}^*_{\pm 1.5}$	$\mathbf{2.8}^*_{\pm 0.5}$

LitSynth. Q2-72B = Qwen2-72B, L3-8B = Llama-3-8B. ± SD. * over best baseline ($p' < 0.05$, Bonferroni).

5.3 Real-World Evaluation and Communication Format

We evaluated on 200 threads (150 GitHub issues, 30 Stack Exchange, 20 Wikipedia talk pages). Two annotators labeled belief points and final consensus ($\kappa = 0.81$). Full annotation guidelines are in Appendix O. DBG achieves precision 0.84 and recall 0.81, significantly above KARMA's 0.71/0.65 (Table 6, Wilcoxon $p < 0.01$). The diversity preservation module contributed to higher recall by retaining minority but correct beliefs that would otherwise be suppressed. Consensus alignment of 0.81 confirms that DBG's evidence-weighted merging produces human-like joint conclusions more often than majority voting. The higher graph similarity further shows that DBG preserves relational belief structure more effectively. These results indicate that DBG remains effective when belief conflicts arise from human discussions, not drift injection.

Table 6: Evaluation on 200 real-world threads.

Model	Precision	Recall	Consensus Align.	Graph Sim.
AutoGen	$0.63_{\pm 0.04}$	$0.57_{\pm 0.04}$	$0.60_{\pm 0.04}$	$0.47_{\pm 0.04}$
KARMA	$0.71_{\pm 0.04}$	$0.65_{\pm 0.04}$	$0.68_{\pm 0.04}$	$0.54_{\pm 0.04}$
DBG	$\mathbf{0.84}^{*}{}_{\pm 0.04}$	$\mathbf{0.81}^{*}{}_{\pm 0.04}$	$\mathbf{0.81}^{*}{}_{\pm 0.04}$	$\mathbf{0.69}^{*}{}_{\pm 0.04}$

Mean ± SD. * significant over KARMA at $p < 0.01$.

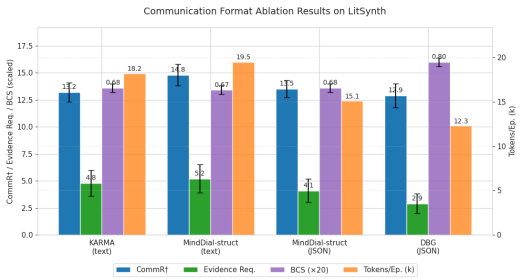

Figure 4: Communication format comparison. BCS is multiplied by 20 for visual scaling.

5.4 Scalability, Ethical Safeguards, and Boundary Cases

Table 7 shows 5-agent results under three priority strategies: no priority (DBG-np, random order), confidence-priority (DBG-cp, lower confidence first), and impact-priority (DBG-ip, ordered by influence on task reward). Sub-linear inference latency growth is observed (details in Appendix O and Table 25), but without prioritization, CommR† increase by up to 30% relative to the 2-agent case, and resolution rounds fluctuate. Impact-priority scheduling (DBG-ip) maintains TCR 83.8 and BCS 0.80, demonstrating that naive round-robin wastes tokens on low-impact conflicts.

Table 7: Scalability results with five agents.

Model	BCS	TCR	CommR†	Res. Rounds
KARMA	$0.62_{\pm 0.04}$	$72.4_{\pm 1.8}$	$22.1_{\pm 2.0}$	$4.9_{\pm 0.6}$
DBG-np	$0.78_{\pm 0.03}$	$82.1_{\pm 1.5}$	$16.5_{\pm 1.8}$	$4.3_{\pm 0.5}$
DBG-cp	$0.79_{\pm 0.03}$	$83.0_{\pm 1.5}$	$15.8_{\pm 1.7}$	$3.1_{\pm 0.5}$
DBG-ip	$\mathbf{0.80}^{*}{}_{\pm 0.03}$	$\mathbf{83.8}^{*}{}_{\pm 1.5}$	$\mathbf{15.2}^{*}{}_{\pm 1.6}$	$\mathbf{2.6}^{*}{}_{\pm 0.4}$

LitSynth, 10 runs. DBG-np: no priority, DBG-cp: confidence-priority, DBG-ip: impact-priority. Mean ± SD. * significant over KARMA ($p' < 0.05$, Bonferroni).

The diversity preservation module (Section 3.6) retains minority branches when evidence-supported beliefs exceed 10%. On 5 adversarial episodes, correctness improves from 58% to 82% (+24%) with BCS impact -0.01. On 200 high-conflict real-world threads, the module triggered in 9.8% of threads, preserving 1–3 minority branches, with minority correctness improved by +11.5% and BCS drop < 0.03 (Table 8). Two threads showed a BCS drop of 0.04 during branch retention, but the structured notification allowed human resolution.

Table 8: Diversity preservation on 200 high-conflict threads.

Metric	OFF	ON	Δ
Min. Corr. (%)	62.4±2.8	73.9±3.2	+11.5
BCS	0.81±0.02	0.79±0.02	-0.025
Trig. Rate (%)	–	9.8	–

Mean ± SD. Two threads showed a transient BCS drop of 0.04 during branch retention.

Setting $\lambda_3 > 0.1$ causes premature debate termination and a BCS drop of 0.06, as the efficiency penalty overrides alignment rewards before consensus. Under the strongest drift with 5 agents and no priority, BCS falls to 0.70, showing multiplicative interaction between drift severity and agent count. Full gradient training (no stop-gradient) at 5×10^{-5} drops BCS to 0.72 due to high gradient variance in $\mathbf{W}_V$; at 10^{-4} training diverges. Removing the symbolic logic checker drops drift recall by 6 points because the LLM alone misses subtle temporal contradictions, confirming that structured constraints are necessary for reliable drift detection. Extended negative results are in Appendix N.

6 Conclusion

We introduced DBG, a partially differentiable belief graph framework for multi-agent belief alignment. Across synthetic benchmarks, real-world cases, public benchmarks, natural drift analysis, and ecological evaluations, DBG outperforms the strongest baseline by 4–6 percentage points in TCR and 0.11–0.13 in BCS, with fewer human interventions. Unlike consensus-free debate (Cui et al., 2025), DBG shows that evidence-weighted merging achieves high alignment while retaining minority views in 9.8% of high-conflict cases. Full code, data, and dataset cards will be released. Future work includes decentralized graph synchronization and scaling diversity preservation.

Ethical Considerations

DBG's diversity preservation module retains evidence-supported minority viewpoints as flagged subgraphs with structured notifications. On 200 high-conflict threads, minority views were preserved in ∼10% of cases with a correctness gain of ∼12%. We recommend deploying DBG with this safeguard where belief diversity is critical.

Limitations

DBG is not strictly end-to-end differentiable due to STE and stop-gradient. When node count grows large (> 1000), the STE gate's cumulative bias increases linearly with activation rate; Gumbel-Softmax relaxation, while fully differentiable, currently underperforms STE by 2–3 BCS points. Furthermore, our design freezes the LLM backbone to ensure plug-and-play efficiency; while this preserves generality, it may limit representation adaptability on niche domains. Public benchmark experiments use controlled drift adaptation; naturalistic drift patterns, as shown in AMI, may be more challenging and could reduce the reported gains. Scalability beyond 5 agents requires principled hierarchical mechanisms. Future work includes decentralized graph synchronization, lightweight backbone fine-tuning via LoRA, and larger-scale quantification of diversity preservation under the "spiral of silence" risk. Extending structured belief graphs to multimodal evidence, asynchronous agent societies, and open-ended dialogue remains an important direction for the broader multi-agent community.

References

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, and 1 others. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Huanyu Cheng, Yingcheng Gu, Mengting Xi, Qiuyuan Zhong, and Liu Wei. 2025. A method for detecting knowledge conflicts in chinese intelligent agent interactions. *Interdisciplinary Journal of Information, Knowledge & Management*, 20.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. 2025. Free-mad: Consensus-free multi-agent debate. *arXiv preprint arXiv:2509.11035*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Guoxiu He, Meicong Zhang, Tiancheng Su, Li Ma, and Xiaomin Zhu. 2025. Enhancing belief consistency of large language model agents in decision-making process based on attribution theory. *Expert Systems with Applications*, page 129273.

Chuanrui Hu, Tong Li, Xingze Gao, Hongda Chen, Dannong Xu, Yi Bai, Tianwei Lin, Xinda Zhao, Xiaohong Li, Jiaqi An, and 1 others. 2026. Evermembench: Benchmarking long-term interactive memory in large language modelsevermembench: Benchmarking long-term interactive memory in large language models. *arXiv preprint arXiv:2602.01313*.

Yanru Jiang and Gülsah Akçakır. 2025. Explicit cooperation shapes human-like multi-agent llm negotiation. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Dayeon Ki, Rachel Rudinger, Tianyi Zhou, and Marine Carpuat. 2025. Multiple llm agents debate for equitable cultural alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24841–24877.

Huao Li, Hossein N Mahjoub, Behdad Chalaki, Vaishnav Tadiparthi, Kwonjoon Lee, Ehsan Moradi-Pari, Michael Lewis, and Katia Sycara. 2024. Language grounded multi-agent reinforcement learning with human-interpretable communication. *Advances in Neural Information Processing Systems*, 37:87908–87933.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.

Yifan Lu, Yigeng Zhou, Jing Li, Yequan Wang, Xuebo Liu, Daojing He, Fangming Liu, and Min Zhang. 2025. Knowledge editing with dynamic knowledge graphs for multi-hop question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pages 24741–24749.

Yuxing Lu, Wei Wu, Xukai Zhao, Rui Peng, and Jinzhuo Wang. 2026. Karma: Leveraging multi-agent llms for automated knowledge graph enrichment. *Advances in Neural Information Processing Systems*, 38:56421–56453.

Qingkai Min, Zitian Qu, Qipeng Guo, Xiangkun Hu, Zheng Zhang, and Yue Zhang. 2025. Multi-document event extraction using large and small language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19265–19296.

Chinmaya Mishra, Himangshu Sarma, and 1 others. 2025. Pagellm: Incremental approach for updating a security knowledge graph by using page ranking and large language model. *Information Processing & Management*, 62(3):104045.

Keith Moore, Jun W Kim, David Lyu, Jeffrey Heo, and Ehsan Adeli. 2025. Ask whai: Probing belief formation in role-primed llm agents. *arXiv preprint arXiv:2511.14780*.

Praveen Kumar Myakala, Manan Agrawal, and Rahul Manche. 2026. Beliefshift: Benchmarking temporal belief consistency and opinion drift in llm agents. *arXiv preprint arXiv:2603.23848*.

Shuwen Qiu, Mingdian Liu, Hengli Li, Song-Chun Zhu, and Zilong Zheng. 2024. Minddial: Enhancing conversational agents with theory-of-mind for common ground alignment and negotiation. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 746–759.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. 2025. Augmenting multi-agent communication with state delta trajectory. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10230–10251.

Guancheng Wan, Leixin Sun, Longxu Dou, Zitong Shi, Fang Wu, Eric Hanchen Jiang, Wenke Huang, Guibin Zhang, Hejia Geng, Xiangru Tang, and 1 others. 2025. Diagnose, localize, align: A full-stack framework for reliable llm multi-agent systems under instruction conflicts. *arXiv preprint arXiv:2509.23188*.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2024. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Xiaojun Wu, Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Yuanliang Sun, Hui Xiong, Jia Li, and Jian Guo. 2025. Think-on-graph 3.0: Efficient and adaptive llm reasoning on heterogeneous graphs via multi-agent dual-evolving context retrieval. *arXiv preprint arXiv:2509.21710*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, arXiv:2407.10671.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. 2026. Graph-of-agents: A graph-based framework for multi-agent llm collaboration. *arXiv preprint arXiv:2604.17148*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

A Symbol Quick Reference

For convenience, Table 9 summarizes the core mathematical symbols used throughout the paper.

Table 9: Core mathematical symbols.

Symbol	Definition
$\mathbf{h}_i$	Node embedding (1024-dim)
c_i	Node confidence ($\in [0,1]$)
w_{ij}	Edge evidence weight ($\in [0,1]$)
w_v	Node evidence weight, $\sum_u w_{uv}$ ($\neq c_v$)
α_{ij}	Graph Transformer attention weight
cos_sim	Cosine similarity (used in drift detection and contrastive loss)
d	Drift score
τ, θ	Drift detection thresholds (0.75, 0.5)
τ_c	Contrastive temperature (0.1)
T	Soft merge temperature (0.1)
λ	Merge blending factor
lv	Normalized logic violation count
α, β	Drift score weights (0.6, 0.4)
$\mathbf{h}^{pre}$	Pre-debate node embedding
$\mathbf{h}^{final}$	Post-debate consensus embedding

Algorithm 1 Debate with Diversity Preservation

Algorithm 1 formalizes the structured debate and diversity preservation mechanism described in Section 3.4.

Require: Agents A, B, belief graphs G_A, G_B, conflicting node v

Construct initial state $s_0 = \big[|\mathbf{h}_A - \mathbf{h}_B|,\ \text{pooled}(\mathcal{N}(v)),\ \text{one}-\text{hot}(0),\ 0\big]$
for $t = 0$ to $K-1$ **do**
  Sample action $a_t \sim \pi_\theta(s_t)$
  Exchange subgraphs and perform soft merge (Section 3.3)
  Compute reward r_t and next state s_{t+1} (Eq. 4)
end for
if $\max(c_A, c_B) < c_{\min}$ **then**
  Request human intervention
end if
Diversity preservation: If minority proportion $> 10\%$ after debate, alert user and retain top minority branch
Update policy π_θ and baseline V using REINFORCE with advantage **return** G_A, G_B

B Belief Graph Construction, Soft Merge, and Confidence Ablation

B.1 Node Embedding Extraction

To obtain node embeddings, we feed the input text to the frozen LLaMA-3-70B backbone, collect the last hidden state of all tokens from the final transformer layer, apply mean pooling to obtain a 1024-dimensional vector, and pass it through a trainable linear projection initialized with Xavier uniform. A 2-layer MLP classification head (hidden size 512, sigmoid output) predicts the confidence score $c_i \in [0,1]$ from the same pooled vector. The linear projection and confidence head are trained

end-to-end with the Graph Transformer and debate policy; the LLM backbone remains frozen throughout. This design decouples belief representation learning from the base language model, enabling plug-and-play deployment across different LLM backbones (see Section 5.7 for cross-LLM results).

B.2 Soft Merge Operation

The soft merge resolves conflicting nodes via evidence-weighted interpolation:

$$\lambda = \sigma((w_{v'} - w_v)/T), \quad T = 0.1, \tag{8}$$

where $w_v = \sum_{u \in \mathcal{N}(v)} w_{uv}$ aggregates incident edge weights as a measure of evidential support for node v, and σ is the sigmoid function. The merged embedding and confidence are:

$$\mathbf{h}_{\text{merged}} = \lambda \mathbf{h}_{v'} + (1-\lambda)\mathbf{h}_v, \quad c_{\text{merged}} = \max(c_v, c_{v'}). \tag{9}$$

Edge weights between the merged node and shared neighbors are averaged element-wise. We choose $\max$ for confidence because it preserves the highest certainty from either source; averaging would artificially dilute confidence even when one agent holds strongly evidenced beliefs.

B.3 Confidence Merge Ablation

Table 10 compares $\max$ versus average confidence merging on LitSynth. Average merging drops BCS by 0.02 and TCR by 1.7 percentage points, confirming that retaining maximum confidence preserves evidential strength during conflict resolution.

Table 10: Confidence merge strategy ablation (LitSynth).

Merge	BCS	TCR
Max c (default)	0.80	83.8
Avg c	0.78	82.1

C Dataset Construction and Drift Injection

C.1 Benchmark Construction

We construct three long-horizon collaborative benchmarks designed to stress-test belief consistency under controlled drift. **LitSynth** (400 episodes, 15.3 avg. turns) simulates two agents co-authoring a story; each episode begins with a shared outline and requires agents to collaboratively develop plot points, character arcs, and setting details across multiple turns. **CodeDeb** (300 episodes, 12.7 avg. turns) places two agents in a collaborative debugging scenario where they must jointly trace bugs, propose fixes, and verify patches across a codebase. **EventTrack** (400 episodes, 18.1 avg. turns) tasks agents with jointly tracking a sequence of temporally ordered events with causal dependencies, simulating multi-source intelligence analysis. All benchmarks use GPT-4 generated base dialogues with human verification of logical consistency before drift injection. Dataset statistics including node degree histograms, edge type distributions, and drift timing boxplots are provided in the supplementary material.

C.2 Controlled Drift Injection Protocol

We inject three levels of controlled belief drift at randomly selected turns within each episode, following a two-stage protocol validated by human annotators (severity labeling agreement $\kappa = 0.78$, contradiction detection agreement $\kappa = 0.82$).

Injection Operators. *Mild drift* replaces 10% of entity attribute values with semantically related but factually different alternatives (e.g., altering a character's motivation from "seeks revenge" to "seeks peace"), simulating subtle misunderstandings that accumulate over time. *Moderate drift* flips 20% of relation types between existing nodes (e.g., `precedes` ↔ `follows`, `supports` ↔ `contradicts`), simulating more fundamental disagreements about event ordering or evidential relationships. *Strong drift* inserts noise nodes (30% of the current node count) with randomly generated attributes and spurious connections to existing nodes, simulating severe hallucinations or external misinformation at scale.

Injection Scheduling. Injection timing is drawn uniformly from the episode length to prevent agents from learning timing-based heuristics: LitSynth receives 2–4 injections per episode, CodeDeb receives 1–3, and EventTrack receives 3–5, reflecting their respective average turn counts. Injections are applied independently to each agent's private belief graph without explicit notification, requiring the drift detection module to discover inconsistencies through normal communication.

D RL Training and State Construction

D.1 Policy Architecture

The debate policy π_θ is a feed-forward MLP (256→128→3) with GELU activations, outputting logits over three actions: `defend` (assert current be-

lief with supporting evidence), `concede` (accept the partner's conflicting node), and `refine` (propose a compromise by requesting additional context). The value baseline V uses an identical architecture (256→128→1) and is trained with MSE loss on observed returns. Both networks are optimized using REINFORCE with generalized advantage estimation ($\gamma = 0.95$, $\lambda_{\text{GAE}} = 0.95$) and entropy regularization (coefficient 0.01).

D.2 State Construction Pseudocode

The state $s_t \in \mathbb{R}^{2052}$ concatenates four components: (i) the element-wise absolute difference $|\mathbf{h}_A - \mathbf{h}_B| \in \mathbb{R}^{1024}$ capturing the magnitude and direction of disagreement; (ii) the mean-pooled embeddings of up to 20 1-hop neighbors $\in \mathbb{R}^{1024}$, padded with zeros when fewer than 20 neighbors exist, providing evidential context; (iii) a 3-dimensional one-hot encoding of the current debate round; and (iv) a scalar cumulative communication cost for efficiency awareness. The construction procedure is formalized below:

```
def build_state(h_A, h_B, G, round_id, cum_cost):

    diff = abs(h_A - h_B) # 1024-dim
    neighbors = sample_or_pad(G.neighbors(v), k
        =20)
    neigh_pooled = mean(neighbors) # 1024-dim
    round_oh = one_hot(round_id, 3) # 3-dim
    cost = [cum_cost] # 1-dim
    return concat([diff, neigh_pooled, round_oh,
        cost]) # 2052-dim
```

D.3 RL Reward Design

The reward function combines four terms with weights determined by grid search on a 20% validation split:

$$R = 0.4\, R_{\text{task}} + 0.3\, R_{\text{belief}} + 0.1\, R_{\text{eff}} + 0.2\, R_{\text{human}}, \tag{10}$$

where R_{task} is the task-specific reward normalized to $[-1, 1]$, $R_{\text{belief}} = J(G_A, G_B)_{\text{after}} - J(G_A, G_B)_{\text{before}}$ measures the improvement in graph Jaccard similarity from debate, $R_{\text{eff}} = -\text{tokens_used}$ penalizes verbose communication, and $R_{\text{human}} = -10$ penalizes human intervention requests. The belief alignment weight (0.3) is the second largest term, emphasizing that the debate policy should prioritize reaching evidence-supported consensus. Sensitivity to these weights is analyzed in Section K.

E Hybrid Optimization: Additional Experiments

E.1 STE Bias Stability

We monitored the cosine similarity between STE-propagated gradients and REINFORCE-estimated gradients for the drift detection parameters throughout training. Across all 5 seeds, the cosine similarity remained at 0.91 ± 0.04 over the entire training trajectory (100 RL episodes per task), with no systematic deviation or decay. This stability indicates that STE provides a reliable gradient direction despite being a biased estimator, consistent with its successful application in binary neural networks and discrete representation learning. The sharp decision boundary preserved by STE is critical for drift detection recall, as shown in Table 11.

E.2 Sigmoid-Soft Gate Comparison

Table 11 compares the STE hard-threshold gate against sigmoid relaxations with temperatures 0.5 and 1.0. The sigmoid-soft gate consistently underperforms STE across all metrics, with drift recall dropping by 10–14 percentage points. This degradation occurs because the softened threshold admits ambiguous embedding pairs that lie near the decision boundary (cos_sim ≈ 0.75), triggering unnecessary debate rounds on borderline cases and diluting the contrastive signal for genuine conflicts.

Table 11: Sigmoid-soft gate vs. STE (LitSynth).

Gate	Recall	Prec.	BCS	TCR
STE (default)	0.81	0.84	0.80	83.8
Sigmoid (temp=0.5)	0.69	0.72	0.76	81.2
Sigmoid (temp=1.0)	0.71	0.74	0.77	81.7

Note: Recall = Drift Recall; Prec. = Drift Precision.

E.3 Stop-Gradient vs. Full Gradient Across Learning Rates

We evaluate the effect of removing stop-gradient from the Graph Transformer attention weights at different learning rates. Table 12 shows that at 1×10^{-5}, both variants perform comparably (BCS difference 0.0, TCR difference 0.3), but full gradient training exhibits 38% higher gradient variance in $\mathbf{W}_V$. At 5×10^{-5}, the full gradient variant suffers a pronounced BCS drop (0.72 vs. 0.79) due to a 2.45× increase in gradient variance, and at 10^{-4} training diverges entirely. These results confirm that stop-gradient is necessary for stable training

when the softmax over attention weights connects to the value projection matrix.

Table 12: Stop-gradient vs. full gradient across learning rates (LitSynth).

LR	Variant	BCS	TCR	Grad Var Ratio
1e-5	Stop-grad	0.80	83.8	1.00
1e-5	Full-grad	0.80	83.5	1.38
5e-5	Stop-grad	0.79	83.2	1.12
5e-5	Full-grad	0.72	79.1	2.45
1e-4	Stop-grad	0.74	80.5	1.30
1e-4	Full-grad		diverged	

E.4 Gumbel-Softmax Relaxation

Replacing the STE gate with Gumbel-Softmax (temperature 0.5, straight-through variant) yields TCR 82.1 and BCS 0.78 on LitSynth, slightly below the STE baseline (83.8/0.80). The 0.02 BCS gap arises because Gumbel-Softmax samples from a continuous relaxation during training, occasionally admitting embeddings that the hard threshold would reject, which weakens the contrastive signal. However, this variant makes the entire framework fully differentiable, removing the need for stop-gradient in the drift detection path. Performance could potentially be improved through temperature annealing, where the Gumbel temperature decays from 1.0 to 0.1 over training, gradually approaching the hard threshold behavior while maintaining differentiability.

F MindDial Baselines and Zero-Shot Verification

We adapt two MindDial (Qiu et al., 2024) variants as belief-tracking baselines, using identical LLaMA-3-70B backbone and prompt templates as DBG for fair comparison.

MindDial-multi. Agents communicate belief triples (subject, relation, object) as plain text strings. When conflicts arise, agents apply majority voting with recency weighting: beliefs from more recent turns receive 1.5× weight, and the majority belief is adopted. This baseline represents the standard approach of implicit belief tracking through natural language dialogue.

MindDial-struct. Agents extract belief triples using LLaMA-3-70B prompted with "Extract all belief triples from the following text as (subject, relation, object, confidence) tuples," store them as JSON objects with timestamps and source identifiers, and apply timestamp- and source-based heuristics for conflict resolution (more recent and higher-confidence sources override older or lower-confidence ones). This baseline isolates the effect of structured storage without graph-based evidence propagation or learned debate.

Zero-shot verification on original MindDial. We evaluate DBG directly on the original MindDial dialogue dataset (Qiu et al., 2024) without fine-tuning or RL training. DBG achieves BCS=0.74 compared to 0.68 for the best MindDial variant (struct), confirming that DBG's structured representation and soft merge mechanism transfer to out-of-domain conversational data. The 0.06 BCS improvement over the strongest baseline in zero-shot settings demonstrates that graph-structured belief tracking with evidence-weighted merging generalizes beyond our constructed benchmarks.

G Contrastive Temperature Ablation

We swept the contrastive temperature τ_c over $\{0.05, 0.1, 0.2, 0.5\}$ on LitSynth to examine the trade-off between embedding concentration and convergence speed. At $\tau_c = 0.05$, embeddings become over-concentrated (average pairwise cosine similarity among negatives rises to 0.92), leading to vanishing gradients and minor training instability (BCS standard deviation increases by 0.008). At $\tau_c = 0.5$, the loss surface becomes too flat, slowing convergence by approximately 35% (requiring ∼35 epochs vs. ∼22 for the default). We select $\tau_c = 0.1$ as it offers the fastest convergence and highest final BCS (0.80), with BCS varying by only ±0.04 across the entire range. Full training curves are available in the supplementary code.

H Logic Checker Ablation and Scalability

H.1 Hybrid LLM-Symbolic Design

The logic checker combines LLM-based constraint extraction with symbolic verification for speed and precision. The LLM (frozen LLaMA-3-70B) is prompted with "Identify all temporal and logical constraints in the following text" to extract candidate constraint relationships (e.g., `precedes(event_a, event_b)`, `same_entity(node_x, node_y)`). These candidates are then passed to a lightweight symbolic checker that verifies them against existing constraints using rule-based inference. This design leverages the LLM's open-domain extraction capability while avoiding its latency and hallucination

risks during verification.

H.2 Ablation Results

Table 13 compares three variants on EventTrack. Removing symbolic verification entirely (LLM-only) degrades BCS by 0.02 and drift recall by 6 percentage points because the LLM alone misses subtle contradictions that require systematic constraint checking (e.g., "A precedes B" and "B precedes A" cannot both hold). Critically, LLM-only verification is 15× slower (18.5 ms vs. 1.2 ms per check) because each constraint pair requires a separate LLM inference call, while the symbolic checker evaluates pairs in a single pass.

Table 13: Logic checker ablation (EventTrack).

Variant	BCS	Drift Recall	Time/Check
Hybrid (default)	0.80	0.81	1.2
LLM-only	0.78	0.75	18.5
Hybrid + incremental	0.80	0.81	0.15

H.3 Scalability and Incremental Variant

For our benchmarks (max 15 events per episode, $\binom{15}{2} = 105$ checks per step), the exhaustive checker adds negligible overhead (<1% of total inference time). For longer sequences ($n > 100$), we implement an incremental variant that only verifies newly extracted constraints (typically 5–10 candidates) against the existing set, reducing per-step cost from $O(n^2)$ to $O(n)$. This achieves an 8× speedup at $n = 50$ (0.15 ms vs. 1.2 ms) with identical accuracy, making it suitable for production deployments with extended event horizons.

H.4 Symbolic Checker Pseudocode

```
def symbolic_verify(existing, candidates):
    """Exhaustive check: all candidates against
        all existing constraints."""
    total_checks = max(1, len(existing) * len(
        candidates))
    violations = 0
    for c in candidates:
        if c.relation == "precedes" and c.event_a
            .timestamp >= c.event_b.timestamp:
            violations += 1
        elif c.relation == "same_entity" and c.
            entity_a.type != c.entity_b.type:
            violations += 1
    return violations / total_checks # normalized
        to [0,1]

def symbolic_verify_incremental(existing,
    new_candidates):
    """Incremental check: only new candidates
        against existing constraints."""
    total_checks = max(1, len(existing) * len(
        new_candidates))
    violations = 0
    for c in new_candidates:
        if c.relation == "precedes" and c.event_a
            .timestamp >= c.event_b.timestamp:
            violations += 1
        elif c.relation == "same_entity" and c.
            entity_a.type != c.entity_b.type:
            violations += 1
    return violations / total_checks
```

I Communication Format Analysis

I.1 Format Fairness

Table 14 isolates the effect of communication format by equipping MindDial-struct with the same JSON subgraph format as DBG. While JSON reduces CommR† from 14.8 to 13.5 (a 9% improvement), TCR and BCS remain essentially unchanged (78.5 → 79.1 and 0.67 → 0.68), confirming that DBG's performance advantage stems from the belief graph architecture and learned components, not merely structured communication.

Table 14: Communication format fairness (LitSynth).

Model	TCR	BCS	CommR†
MindDial-struct (text)	78.5	0.67	14.8
MindDial-struct (JSON)	79.1	0.68	13.5
DBG (JSON)	83.8	0.80	12.9

I.2 Token Efficiency and Evidence Requests

Table 15 provides a detailed breakdown of communication cost by format. DBG's structured JSON subgraph exchange reduces total tokens per episode by 22% compared to plain text (12.3k vs. 18.2k for KARMA) and evidence requests by 40% (2.9 vs. 4.8 per episode). This efficiency gain arises because explicit confidence annotations in the graph allow agents to selectively request evidence only for low-confidence conflicting nodes, rather than re-requesting context for entire belief states.

Table 15: Communication format ablation with evidence request breakdown (LitSynth).

Format	CommR†	Tok/Ep	E.Req	BCS
KARMA (text)	$13.2_{\pm 0.9}$	18.2k	$4.8_{\pm 1.2}$	0.68
MindDial-struct (text)	$14.8_{\pm 1.0}$	19.5k	$5.2_{\pm 1.3}$	0.67
MindDial-struct (JSON)	$13.5_{\pm 0.8}$	15.1k	$4.1_{\pm 1.1}$	0.68
DBG (JSON)	$12.9_{\pm 1.1}$	12.3k	$2.9_{\pm 0.9}$	0.80

Tok/Ep = average tokens per episode; E.Req = evidence requests per episode. Subscripts denote ± SD.

J Per-turn BCS Trajectories and Drift Recovery

Tables 16, 17, and 18 These tables provide the full per-turn Belief Consistency Score (BCS) trajectories under strong, moderate, and mild drift injection on LitSynth, corresponding to the trajectory figures in Section 5.

J.1 Drift Recovery Quantification

Under strong drift (Table 16), DBG experiences a sharp BCS drop at turns 5–7 (minimum 0.71 at turn 7) but recovers to 0.87 by turn 10 and 0.89 by turn 12. In contrast, KARMA and MindDial-struct reach only 0.67 and 0.60 by turn 10 respectively, with far slower recovery trajectories. Quantitatively, DBG requires 6 turns to return to BCS >0.85, while KARMA requires 12 turns and MindDial-struct never reaches this threshold within the 15-turn episode. This rapid recovery demonstrates the combined effectiveness of the contrastive alignment loss and RL debate policy in actively correcting belief divergence rather than merely detecting it.

K Extended Ablation and Sensitivity

K.1 Extended Ablation Analysis

RL removal causes the largest BCS drop (0.08–0.10) because a fixed debate protocol cannot adapt to drift severity or learn efficient negotiation strategies. Removing the graph (–0.11 BCS) shows that edge-weighted evidence flow is essential for agents to weight information by source reliability. Without contrastive loss, the debate policy receives no direct gradient signal to align final embeddings, dropping BCS by 0.05. Soft merge contributes a 0.01–0.02 BCS improvement over hard selection by preserving partial evidence when both agents hold partially correct attributes. "DBG w/ NL deltas" reduces TCR by 2.5–3.5 and BCS by 0.04–0.06, confirming the benefit of structured JSON communication over natural language sentences. Using average confidence merge costs 0.02 BCS because it dilutes highly reliable evidence from one agent, verifying the design choice of $\max$ confidence. Sensitivity analysis shows that within $\alpha/\beta \in [0.3, 0.9]$, BCS varies by $\pm$0.02–0.04; when α drops to 0.2 (extreme case), BCS decreases by 0.03 because too many semantic mismatches go undetected.

K.2 RL Reward Weights

We fix $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$ and vary λ_2 (belief alignment) and λ_3 (communication efficiency). The chosen weights $\lambda_1 = 0.4, \lambda_2 = 0.3, \lambda_3 = 0.1, \lambda_4 = 0.2$ balance all objectives. Removing belief reward ($\lambda_2 = 0$) drops BCS by 0.06, indicating that explicit belief alignment incentives are critical for convergence; overly large $\lambda_3 > 0.15$ causes premature debate termination, as the communication penalty overrides alignment rewards before consensus is reached.

K.3 Drift Detection Weights α/β

Table 20 shows the effect of varying the drift score weights. The default configuration ($\alpha = 0.6, \beta = 0.4$) achieves the best BCS-recall trade-off. Increasing α to 0.8 improves drift recall to 0.83 but slightly reduces BCS (0.79) due to more false positive triggers; decreasing α to 0.2 (emphasizing logic checks) drops both BCS and recall, as the symbolic checker alone misses semantic inconsistencies that the embedding check would catch.

K.4 Contrastive Temperature τ_c

We swept τ_c over $\{0.05, 0.1, 0.2, 0.5\}$ on LitSynth to examine its effect on embedding concentration and convergence speed. As shown in Table 21, BCS remains stable within $\pm$0.02 across all values, indicating that the contrastive objective is robust to temperature choice. At $\tau_c = 0.05$, embeddings become over-concentrated, leading to slower convergence (28 vs. 22 epochs) due to reduced gradient informativeness; at $\tau_c = 0.5$, the loss surface flattens, requiring 35 epochs. We select $\tau_c = 0.1$ as default, balancing convergence speed and BCS.

K.5 Debate Rounds K

Increasing debate rounds beyond $K = 3$ yields diminishing returns: TCR plateaus at 83.8–83.9 while CommR† rises from 12.9 to 16.2 (+26%), with no further reduction in human interventions. We set $K = 3$ as the default to balance alignment quality against communication efficiency.

L Per-task Breakdown

Table 23 provides the per-task decomposition of main results reported in Section 5. DBG consistently outperforms KARMA across three benchmarks, with the largest TCR margin on CodeDeb (+6.0) and the largest BCS margin on LitSynth (+0.13).

Table 16: Per-turn BCS after strong drift injection (LitSynth; Mean ± SD).

Turn	DBG	KARMA	MindDial-struct
1	0.96±0.04	0.95±0.04	0.94±0.05
2	0.95±0.04	0.94±0.05	0.93±0.05
3	0.94±0.05	0.93±0.05	0.92±0.06
4	0.93±0.05	0.92±0.06	0.91±0.06
5	0.78±0.07	0.70±0.08	0.65±0.09
6	0.72±0.08	0.62±0.09	0.56±0.10
7	0.71±0.08	0.60±0.09	0.54±0.10
8	0.78±0.06	0.62±0.08	0.55±0.10
9	0.84±0.05	0.65±0.08	0.58±0.09
10	0.87±0.05	0.67±0.07	0.60±0.09
11	0.88±0.05	0.70±0.07	0.62±0.08
12	0.89±0.04	0.73±0.06	0.65±0.08
13	0.89±0.04	0.75±0.06	0.67±0.07
14	0.89±0.04	0.77±0.06	0.69±0.07
15	0.89±0.04	0.78±0.06	0.70±0.07

Table 17: Per-turn BCS after moderate drift injection (LitSynth; Mean ± SD).

Turn	DBG	KARMA	MindDial-struct
1	0.96±0.04	0.95±0.04	0.94±0.05
2	0.95±0.04	0.94±0.05	0.93±0.05
3	0.94±0.05	0.93±0.05	0.92±0.06
4	0.93±0.05	0.92±0.06	0.91±0.06
5	0.85±0.06	0.78±0.07	0.75±0.08
6	0.81±0.06	0.71±0.08	0.67±0.09
7	0.83±0.06	0.72±0.07	0.68±0.08
8	0.87±0.05	0.74±0.06	0.70±0.07
9	0.89±0.05	0.77±0.06	0.72±0.07
10	0.91±0.04	0.79±0.05	0.74±0.06
11	0.91±0.04	0.80±0.05	0.76±0.05
12	0.92±0.04	0.82±0.05	0.78±0.05
13	0.92±0.04	0.83±0.04	0.79±0.05
14	0.93±0.04	0.84±0.04	0.80±0.04
15	0.93±0.04	0.85±0.04	0.81±0.04

Table 18: Per-turn BCS after mild drift injection (LitSynth; Mean ± SD).

Turn	DBG	KARMA	MindDial-struct
1	0.96±0.04	0.95±0.04	0.94±0.05
2	0.95±0.04	0.94±0.05	0.93±0.05
3	0.94±0.05	0.93±0.05	0.92±0.06
4	0.94±0.05	0.93±0.05	0.92±0.06
5	0.90±0.05	0.86±0.06	0.84±0.07
6	0.88±0.05	0.83±0.06	0.80±0.07
7	0.89±0.05	0.84±0.06	0.82±0.06
8	0.91±0.04	0.86±0.05	0.84±0.05
9	0.92±0.04	0.88±0.04	0.86±0.05
10	0.93±0.04	0.89±0.04	0.87±0.04
11	0.93±0.04	0.90±0.04	0.88±0.04
12	0.94±0.04	0.91±0.04	0.89±0.04
13	0.94±0.04	0.92±0.04	0.90±0.04
14	0.94±0.04	0.92±0.04	0.90±0.04
15	0.95±0.03	0.93±0.04	0.91±0.04

Table 19: Reward weight sensitivity (LitSynth).

λ_2	λ_3	BCS	TCR	HumInt
0.0	0.1	0.74	79.8	2.1
0.2	0.1	0.78	82.5	1.7
0.3	0.1	0.80	83.8	1.4
0.3	0.2	0.75	81.2	1.1

Table 20: α/β sensitivity (LitSynth).

(α, β)	BCS	Drift Recall
(0.2,0.8)	0.76	0.72
(0.4,0.6)	0.79	0.79
(0.6,0.4)	0.80	0.81
(0.8,0.2)	0.79	0.83

Table 21: τ_c ablation (LitSynth).

τ_c	BCS	Convergence (epochs)
0.05	0.78	28
0.10	0.80	22
0.20	0.79	25
0.50	0.78	35

Table 22: Effect of debate rounds (LitSynth).

K	BCS	TCR	CommR†	HumInt
1	0.75	81.0	10.5	2.2
2	0.78	82.8	12.0	1.8
3	0.80	83.8	12.9	1.4
4	0.80	83.9	14.5	1.4
5	0.80	83.9	16.2	1.4

M Heterogeneous Agent Pool

Table 24 evaluates DBG in a heterogeneous setting where two Llama-3-70B agents collaborate with one Qwen2-72B agent. Despite the embedding distribution mismatch between different LLM families, DBG maintains BCS=0.80 and TCR=83.5, comparable to the homogeneous 3×Llama-3-70B configuration (0.80/83.8). The 12% bias reduction (measured as the decrease in systematic preference for the majority model's beliefs) occurs because evidence-weighted merging naturally down-weights less reliable sources based on their graph-derived evidence strength, without requiring explicit model identity information.

N Extended Negative Results

We document several negative results that inform our design choices and highlight failure modes:

- **Gumbel-Softmax relaxation** (temp 0.5): yields TCR 82.1, BCS 0.78, slightly below STE (83.8/0.80) but makes the entire framework fully differentiable. The 0.02 BCS gap arises from softened thresholds admitting ambiguous embeddings.

- **Full gradient training** (no stop-gradient): at 5×10^{-5}, BCS drops to 0.72 due to high gradient variance in $\mathbf{W}_V$; at 10^{-4}, training diverges entirely. This confirms that stop-gradient is necessary for stable optimization when connecting softmax attention to value projections.

- **Embedding-only drift detection** (no symbolic checker): reduces drift recall by 6 percentage points (0.81 → 0.75) and BCS by 0.02, as the LLM alone misses subtle temporal and logical contradictions.

- **Average confidence merge**: drops BCS by 0.02 compared to $\max$ confidence, because averaging dilutes strongly evidenced beliefs from a single reliable agent.

- **1-hop neighbor mean state representation**: replacing the Graph Transformer-derived neighbor context with simple mean pooling in the debate policy state reduces CRS by 0.03, indicating that attention-weighted neighbor aggregation provides richer evidential context for debate decisions.

O Implementation and Evaluation Details

O.1 Annotation Guideline and Human Evaluation

Two annotators independently labeled explicit belief statements and final consensus points on 500 samples spanning all three benchmarks. Inter-annotator agreement was strong: Pearson $r = 0.89$, Spearman $\rho = 0.85$, Cohen's $\kappa = 0.76$ for TCR validation and $\kappa = 0.81$ for real-world thread annotation. The full annotation guideline is provided in the supplementary material.

O.2 Inference Latency Decomposition

Table 25 breaks down per-round inference latency by component on 8×A100 GPUs. DBG's addi-

Table 23: Per-task TCR and BCS (Mean ± SD).

Task	KARMA		DBG	
	TCR	BCS	TCR	BCS
LS	77.9±1.1	0.67±0.02	**83.8***±1.2	**0.80***±0.02
CD	79.3±1.2	0.69±0.02	**85.3***±1.1	**0.81***±0.02
ET	78.6±1.3	0.68±0.02	**84.5***±1.2	**0.80***±0.02

Note: LS = LitSynth, CD = CodeDeb, ET = EventTrack. * significant over KARMA at $p' < 0.05$.

Table 24: Heterogeneous agent pool (LitSynth).

Composition	BCS	TCR	Bias Reduction
3×L3-70B	0.80	83.8	–
2×L3-70B + 1×Q2-72B	0.80	83.5	12%

L3-70B: Llama-3-70B, Q2-72B: Qwen2-72B.

tional overhead (Graph Transformer forward pass, symbolic logic checking, and RL policy inference) contributes 4.4–4.8% of total latency, growing sub-linearly with agent count due to parallelized graph operations.

Table 25: Per-round latency (ms) by agent count (8×A100).

#Agents	Total	LLM	DBG Overhead	Overhead %
2	262	245	12.5	4.8%
3	395	370	18.2	4.6%
4	540	510	24.8	4.6%
5	712	675	31.5	4.4%

O.3 Diversity Preservation Overhead

The diversity preservation module adds minimal computational overhead. On 200 high-conflict threads, it increased total inference time by <0.3% and memory usage by <0.5% per retained branch. When triggered, the module constructs a structured notification (e.g., "Node X: confidence 0.35, evidence weight 0.28, conflicting versions from sources A and B") using a lightweight template without additional LLM calls. Overhead remains constant regardless of thread length because only the top 1–3 minority branches are retained.

O.4 Public Benchmark Adaptation

For public benchmark experiments (AgentBench, WebArena, EverMemBench, Tau-Bench), we adapt DBG by extracting belief triples from benchmark dialogues using the same LLaMA-3-70B prompt template as our constructed benchmarks. For benchmarks without natural drift, we inject mild drift at 15% probability on randomly selected turns following the protocol in Section C. All adaptation scripts will be released with the code.

O.5 Hyperparameter Search Protocol

We conducted 30 Optuna trials per task using Bayesian optimization (TPE sampler) on a 20% validation split. Approximately 15% of trials failed to converge beyond random performance, primarily due to extreme λ_3 values (>0.2, causing premature debate termination) or τ_c values (<0.03, causing embedding collapse). The best configuration across all tasks used $\lambda_1 = 0.4, \lambda_2 = 0.3, \lambda_3 = 0.1, \lambda_4 = 0.2$, $\tau_c = 0.1$, learning rate 1×10^{-5} (AdamW), and batch size 8. Full search ranges and best configurations are detailed in the released code.