SEMANTIC COOPERATIVE GAMES FOR CONTRIBUTION ATTRIBUTION IN LLM-BASED MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based multi-agent systems produce outputs through ordered interactions, yet contribution attribution lacks a unified semantic account of how agents generate, preserve, and transform task-relevant information. Existing methods typically rely on counterfactual valuation by removing agents or comparing system scores across agent subsets. These approaches require costly workflow reruns that may alter downstream context. They also leave much of the intermediate semantic information unused, limiting their ability to explain specific contributions within a realized trajectory.

We propose Semantic Cooperative Games (SCG), a framework that represents a realized language interaction trajectory as a semantic generation hypergraph and derives an agent-level value function from its semantic support relations. We define the Semantic Shapley Value (SSV) and introduce SLIC, a single-trajectory algorithm that constructs the hypergraph, recovers minimal semantic supports, and applies Boolean absorption to compute SSV without rerunning agent subsets.

Our theory provides an axiomatic foundation for SSV and guarantees that it reduces to the classical Shapley value whenever the latter is well-defined within our framework. In a three-agent medical question-answering workflow where the classical Shapley value is well-defined, SLIC reduces API calls by 85.7% relative to an exhaustive Shapley baseline, achieving a Kendall rank correlation of $\tau_b = 0.814$ with the baseline attribution. In more general workflows involving multiple roles, SSV correlates strongly with score drops under moderate to strong perturbations, while revealing cases in which semantic contribution and failure impact diverge. SCG provides an interpretable paradigm for contribution attribution and lays the groundwork for systematic annotation of semantic actions and their contributions in Group-of-Thought workflows.

Code is available at https://anonymous.4open.science/r/SCG_SLIC-D3A4.

1 INTRODUCTION

Large language model (LLM)-based multi-agent systems increasingly solve complex tasks by distributing computation across interacting agents with specialized roles, communication protocols, and explicit workflow structures. Frameworks such as CAMEL (Li et al., 2023a), AutoGen (Wu et al., 2023), MetaGPT (Hong et al., 2024), and MedAgents (Tang et al., 2024) demonstrate the value of this organization for task decomposition, deliberative reasoning, and domain-specific collaboration. However, system behavior is no longer determined by a single model response. It emerges from sequences of agent actions, message dependencies, and intermediate semantic transformations. This makes it difficult to determine which agents shape the final output and where failures originate. This motivates the *contribution attribution problem* (CAP): given a realized workflow trace, a value-reading interface, and a set of participating agents, assign each agent a contribution score that explains its role in producing the observed output. Such attribution supports failure diagnosis (Zhang et al., 2025b), workflow comparison (Yang et al., 2025), agent pruning (Zhang et al., 2025a), and robustness analysis (Chan et al., 2024). CAP is also a concrete instance of the broader credit assignment problem, including temporal credit assignment in reinforcement learning (Sutton, 1984) and structural credit assignment in multi-agent systems (Agogino & Tumer, 2004).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A broad class of attribution methods can be characterized as a *counterfactual valuation* route. Instead of identifying contribution from the realized workflow itself, these methods estimate how the system value changes under hypothetical modifications. They construct perturbed executions by removing, replacing, or masking an agent, message, feature, action, or workflow component (Ribeiro et al., 2016; Zeiler & Fergus, 2014; Lundberg & Lee, 2017). The perturbed system is then sampled or rerun, and attribution is computed from the resulting value differences:

$$\Delta\text{System} \xrightarrow{\text{sample / rerun}} \widehat{\text{Value}} \xrightarrow{\text{value difference}} \text{Attribution}.$$

Here, $\widehat{\text{Value}}$ denotes a value estimate obtained from altered executions rather than from the original trace. This counterfactual valuation route spans multi-agent reinforcement learning, cooperative game theory, and LLM-agent workflow attribution. COMA estimates agent-specific counterfactual action values with a centralized critic (Foerster et al., 2018), and difference-reward methods compare global rewards with counterfactual rewards obtained by removing or clamping an agent's contribution (Wolpert & Tumer, 2001; Nguyen et al., 2018). In LLM-based collaboration, C3 estimates message-level causal impact through fixed-continuation replay and leave-one-out interventions (Chen et al., 2026), while ShapleyFlow evaluates alternative agentic workflow configurations (Yang et al., 2025). More generally, cooperative game-theoretic methods allocate contribution through marginal comparisons over coalitions or workflow components (Shapley, 1953; Aumann & Shapley, 1974; Young, 1985). Despite their different intervention targets and aggregation rules, these methods share a common dependency: attribution is mediated by scalar values assigned to hypothetical configurations. This dependency creates two obstacles in language-mediated workflows as illustrated in Fig. 1.

Problem 1: Contribution is workflow-dependent. An LLM workflow induces a partial-order dependency structure over agent actions and messages. A set-based counterfactual, such as keeping or removing a subset of agents or components, therefore does not fully define the attribution target. The same subset can have different effects under different execution orders, routing rules, or downstream dependencies. An agent's contribution depends not only on what it generates, but also on where that content appears and how it is used downstream.

Problem 2: Rerun-based valuation is costly and unstable. When values are obtained through rerun, replay, or rollout, each intervention can rewrite downstream context and trigger different model responses. This adds model calls, stochastic variance, and sensitivity to decoding, prompting, and orchestration details. Even with a surrogate evaluator, attribution remains tied to evaluator stability. These difficulties grow rapidly with the number of agents, messages, components, and interventions.

These obstacles suggest that CAP should be treated not only as a value-comparison problem, but also as a workflow-structural problem. In LLM-based multi-agent systems, contribution is expressed through the realized trajectory, where agents produce intermediate content, pass information downstream, reuse prior messages, and transform semantic states. Since black-box LLM internals are difficult to inspect, we use the language trajectory as the observable layer of this information flow. Prior work on agent-level monitoring and process evaluation (Chan et al., 2024; Gritta et al., 2026) shows that such trajectories contain useful process signals. For CAP, the relevant signals are semantic: agents introduce, preserve, rewrite, or combine task-relevant content that later agents may reuse in reasoning. This motivates a shift from workflow structure to *language flow*, which exposes the semantic states and transformations through which task value is produced.

To make this semantic dependency structure analyzable, we propose **Semantic Cooperative Games (SCG)**. As shown in Fig. 1, SCG converts a realized language flow into a semantic generation hypergraph. The graph contains semantic nodes, support links, an ownership map from links to agents, and weighted output nodes read by a rubric. Given the player set N and graph G, SCG defines a semantic game representation (N, G). The graph then induces an agent-level semantic value function, making workflow dependency part of the attribution object. On this induced value function, we define the **Semantic Shapley Value (SSV)**. The support layer uses Boolean logic to represent sufficient semantic supports, following Boolean games and cooperative Boolean games (Harrenstein et al., 2001; Dunne et al., 2008). The allocation layer follows Shapley-style axiomatic allocation (Shapley, 1953; Young, 1985). Thus, SSV extends the classical Shapley value from unordered coalitions to semantic support structures. Under standard set-based, fully observable, and no-order-dependence conditions, it coincides with the classical Shapley value.

To compute SSV from a single trajectory, we propose **SLIC** (Semantic Logic Inversion for Contribution). SLIC reads the value-bearing output nodes and their rubric weights, recursively traces

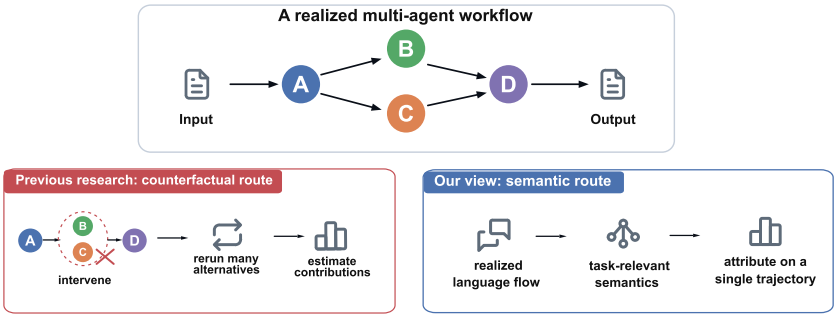


Figure 1: Counterfactual vs. semantic routes for contribution attribution. *Bottom left:* The traditional counterfactual route estimates contributions by masking components and rerunning the workflow to measure value changes, requiring many model calls and high computational cost. *Bottom right:* The semantic route analyzes a single realized execution by tracing the language/semantic flow across agents. Contributions are computed directly from this dependency structure via the Semantic Shapley Value, with no reruns.

semantic predecessors in language flow, and constructs G. It then extracts minimal link-level supports, maps links to agents through ownership, applies Boolean absorption, and aggregates multilinear coefficients to obtain SSV. This avoids rerunning agent subsets. In a medical benchmark satisfying the reduction conditions, SLIC matches a Monte Carlo baseline while reducing computation cost by 93.3%. In more general multi-role workflows, SSV tracks agent-level score drops under perturbations and exposes cases where limited semantic contribution can still carry large failure impact.

Our contributions are as follows:

- We propose **SCG**, a semantic cooperative-game framework that represents a realized language flow as a semantic generation hypergraph and induces an agent-level value function from workflow-dependent semantic supports.
- We define the **Semantic Shapley Value (SSV)** and introduce a single-trajectory algorithm that computes SSV without counterfactual reruns. We prove that SSV coincides with the classical Shapley value under standard coalition-game conditions.
- We validate SCG/SLIC on reduction-consistency and structural-diagnosis benchmarks, showing that SLIC reduces computation cost while preserving Shapley consistency and that SSV provides interpretable attribution in complex multi-role workflows.

2 Semantic Cooperative Games

To define a Shapley-style attribution game for an LLM workflow, we first specify the semantic object from which value will be induced. An agent is observed through its language action: it produces, carries forward, or transforms semantic content. We therefore model each agent action as a semantic transformation and analyze the workflow through its dual language flow.

Formally, let $T = (\mathcal{L}, H)$, where each state in $\mathcal{L} \subseteq X$ is a semantic state produced at some step, and $H = \{h_u\}$ is the set of agent actions. To connect this trace to value, we use the rubric or value-reading interface. In rubric-based evaluation, the final scalar score decomposes into several score-bearing output semantics. Let $O = \{o_1, \ldots, o_K\}$ denote these output semantics, and let ω assign their weights. The goal of this section is to extract the value-relevant semantic nodes from language flow and use them to construct a semantic generation hypergraph.

Assumption 1 (Semantic Separability and Source Identifiability). For any execution position u, every semantic node $s \subseteq y_u$ relevant to this paper is one of three types: native semantics, derived semantics, or persistent semantics. If s is derived or persistent, its prior semantic source in the input state x_u can be identified.

Under Assumption 1, each agent action h_u gives a direct predecessor set for every semantic node it produces:

$$\mathrm{src}_{h_u}(s) = \begin{cases} \emptyset, & s \text{ is native,} \\ \{\tilde{s}\}, & s \text{ is persistent,} \\ P,\ P \subseteq x_u, & s \text{ is derived,} \end{cases}$$

where $\mathrm{src}_{h_u}(s)$ is the direct predecessor set of semantic node s under action h_u. For each semantic node, let $h(s)$ be the action that produced it. To recover the feature space of the game, we recursively trace each output node $o_k \in O$ back to its predecessors. This yields the set of value-relevant semantic nodes S. We define the initial semantic nodes as $I := \{s \in S : \mathrm{src}_{h(s)}(s) = \emptyset\}$. For each $s \in S \setminus I$, the directed hyperedge $e_s : \mathrm{src}_{h(s)}(s) \to s$. This gives the semantic generation hypergraph $G = (S, L, N; \alpha, \omega, O, I)$, where N is the agent set in the current trajectory, $L := \{e_s : s \in S \setminus I\}$, and $\alpha : L \to N$ records which agent owns each link, $O \subseteq S$ denotes the output semantic nodes at which task value is evaluated, and $\omega : O \to \mathbb{R}$ assigns their value weights.

We construct G in this backward direction because attribution should focus on the system behaviors expressed through task-relevant semantics. Starting from O anchors the construction at the semantic features read by the value interface, while tracing predecessors identifies how the system produced, reused, or transformed those features. The resulting graph is therefore a compact description of the realized behaviors from which agent-level support logic and value can be induced. Additional formal details and examples are deferred to the Appendix I.1. Therefore, we introduce the definition of Semantic Cooperative Game as follows.

Definition 1 (Semantic Cooperative Game)**.** Given a language flow and its induced semantic generation hypergraph $G = (S, L, N; \alpha, \omega, O, I)$, we define the semantic game representation as $\mathcal{G}(G) := (N, G)$. Here, N is the player set, and G is the semantic generation hypergraph from which an agent-level semantic value function is induced. Players participate through their owned hyperedges, and value is read from the weighted output nodes O.

Remark 1 (Identifying and validating semantic hypergraphs)**.** For linguistic expressions z, z', let $c(z, z') \in \{0, 1\}$ indicate semantic equivalence, with 1 denoting agreement. Expressions judged equivalent are grouped into the same semantic node. **One possible criterion** for semantic support is

$$p_u(s \mid x_u) - p_u(s \mid x_u \setminus P) \geq \delta,$$

where $P \subseteq x_u$ is a candidate predecessor set, $p_u(s \mid x)$ is the probability of generating s from x under a fixed agent configuration, and $\delta > 0$ specifies a substantial decrease. Alternative sources require joint interventions. Persistence retains predecessor semantics, derivation transforms or combines them, and native semantics have no predecessor source.

These distinctions depend on the discriminator's semantic judgments; we therefore do not claim a unique ground-truth semantic hypergraph. Given G, however, our allocation axioms uniquely determine the Semantic Shapley Value. We posit that an LLM can identify these simple local causal or inferential relations without performing counterfactual tests for each link. We therefore use an LLM as the discriminator and provide counterfactual validation of its judgments in Appendix E. Although different analyzer produce different hypergraphs, the resulting allocations remain similar in our experiments; see Appendix F.

3 SLIC: Semantic Logic Inversion for Contribution

Once G is available, it can induce many possible value functionals and attribution rules, depending on which semantic outputs are selected and how they are valued. In this work, we focus on one canonical allocation rule on G: the Semantic Shapley Value (SSV), which is uniquely determined by the Shapley axioms over G. To compute SSV, SLIC recovers the support logic of value-bearing output nodes to generate G from the realized trajectory,and applies Boolean absorption and multilinear allocation to compute SSV. This section formalizes this single-trajectory computation, following the overall pipeline illustrated in Fig. 2.

3.1 From Semantic Supports to SSV

Algorithm 1 constructs an attribution game from a realized trajectory in two stages. First, SLIC traces backward from the output nodes O to recover the value-relevant semantic flow and construct the

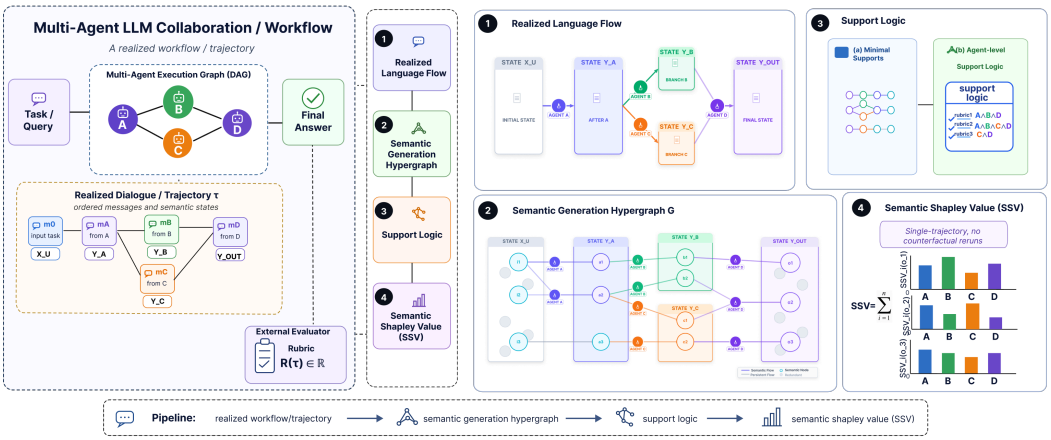


Figure 2: **Overall SCG/SLIC pipeline for semantic attribution.** The key idea is to compute agent contributions directly from a *single realized execution* by leveraging its semantic structure. A workflow is transformed into a semantic generation hypergraph G, from which SLIC traces value dependencies, recovers minimal support logic, and computes the Semantic Shapley Value (SSV). This enables *single-pass, structure-based attribution* without counterfactual reruns.

Algorithm 1 SLIC: construct G from language flow and compute SSV

Require: realized trajectory τ, value-reading interface R, ownership map α
Ensure: semantic generation hypergraph G, support logic, and SSV

Read $O = \{o_1, \ldots, o_K\}$ and its weights ω from R
for $k = 1, \ldots, K$ **do**
    Starting from o_k, recursively trace back its direct predecessors $\mathrm{src}_{h(s)}(s)$ along language flow until no new semantic nodes appear
end for
Let the set of all traced semantic nodes be S, and define $I = \{s \in S : \mathrm{src}_{h(s)}(s) = \emptyset\}$
For each $s \in S \setminus I$, define $e_s : \mathrm{src}_{h(s)}(s) \to s$, and let $L = \{e_s : s \in S \setminus I\}$
Construct $G = (S, L, N; \alpha, \omega, O, I)$
for $k = 1, \ldots, K$ **do**
    Extract the minimal link-level supports of o_k, lift them to the agent level, and apply Boolean absorption to obtain the support logic f_{o_k}
    Expand f_{o_k} into multilinear / Möbius coefficients $c_T^{(o_k)}$
end for
Aggregate the weighted coefficients over all k and output $\mathrm{SSV}(G)$

semantic generation hypergraph G. Second, it converts this graph to the agent level by identifying which subsets of agents provide sufficient semantic support for each value-bearing output node.

Fix an output node $o_k \in O$. Let $\mathcal{C}_{o_k} \subseteq 2^L \setminus \{\emptyset\}$ denote its minimal link-level conjunctive supports. Each $C \in \mathcal{C}_{o_k}$ is a set of semantic links whose joint presence is sufficient to support o_k. Through the ownership map, each link support C induces the agent set $N(C) := \{\alpha(\ell) : \ell \in C\} \subseteq N$. After deduplication and Boolean absorption, these induced agent sets yield the minimal support family $\mathcal{M}_{o_k}$. The corresponding Boolean support function is

$$f_{o_k}(x) = \bigvee_{M \in \mathcal{M}_{o_k}} \bigwedge_{i \in M} x_i, \qquad x \in \{0,1\}^{|N|}. \tag{1}$$

Here, $x_i = 1$ indicates that agent i is included in the selected agent configuration. Thus, $f_{o_k}(x) = 1$ if and only if the selected agents semantically support the rubric node o_k. Aggregating these support functions with rubric weights yields the agent-level semantic value function induced by G:

$$V_G^{\mathrm{sem}}(x) := \sum_{o_k \in O:\, x_{o_k}(G)=1} \omega(o_k) f_{o_k}(x). \tag{2}$$

This equation is the key step that separates SLIC from rerun-based attribution. The value function is not queried from counterfactual executions. Instead, it is induced analytically from the semantic supports observed in the realized trajectory.

To allocate this induced value, we expand each support function into its unique multilinear, or Möbius, form:

$$f_{o_k}(x) = \sum_{T \subseteq N} c_T^{(o_k)} \prod_{i \in T} x_i, \qquad c_T^{(o_k)} = \sum_{R \subseteq T} (-1)^{|T|-|R|} f_{o_k}(\mathbf{1}_R). \tag{3}$$

The coefficient $c_T^{(o_k)}$ measures the semantic interaction carried jointly by the agent set T for the output node o_k. These coefficients form the basis on which Shapley-style allocation is performed in Definition 2.

Definition 2 (Semantic Shapley Value)**.** Given the semantic representation of the game $\mathcal{G}(G) = (N, G)$, the semantic Shapley value induced by G is

$$\mathrm{SSV}_i(G) := \sum_{o_k \in O : x_{o_k}(G)=1} \omega(o_k) \sum_{T \subseteq N :\, i \in T} \frac{c_T^{(o_k)}}{|T|}, \qquad i \in N.$$

3.2 Axiomatic Interpretation

Definition 2 applies Shapley-style allocation to the semantic support expansion induced by G. Each coefficient $c_T^{(o_k)}$ is a semantic support term jointly carried by the agents in T. The allocation of such a term follows the Shapley axioms: **efficiency** allocates the whole coefficient, **dummy** gives zero to agents outside T, and additivity sums allocations over all semantic terms. **Symmetry** gives equal shares to agents that play the same role inside the same semantic support term (Shapley, 1953; Young, 1985). Hence each $i \in T$ receives $c_T^{(o_k)}/|T|$. In this sense, SSV keeps the core Shapley behavior of sharing joint contribution, while avoiding counterfactual coalition evaluation.

Boolean absorption acts before allocation and handles a different kind of redundancy. It removes logically redundant support expansions so that equivalent support expressions define the same f_{o_k}, the same coefficients, and the same SSV. For example, $(A \wedge B) \vee B \equiv B$ removes the redundant A-path before Shapley allocation. By contrast, if a semantic term genuinely requires the joint support $A \wedge B$, absorption does not remove either agent; symmetry splits the remaining term between them. Supplementary link-support and absorption examples are given in Appendix J.1–J.2; the degeneration of LOO under redundancy is discussed in Appendix L.1. The full derivation is given in Appendix G.

Theorem 1 (Equivalence of Semantic Shapley Value with the Classical Shapley Value)**.** *Suppose the system satisfies set-based value reachability, full observability, and no masking or order dependence along the realized language flow. Let $v : 2^N \to \mathbb{R}$ be a classical coalition value function, and let*

$$\phi_i(v) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!(|N| - |C| - 1)!}{|N|!} \big[v(C \cup \{i\}) - v(C)\big] \tag{4}$$

denote the classical Shapley value.

Let $V_G^{\mathrm{sem}} : 2^N \to \mathbb{R}$ be the coalition function induced by the semantic structure of the system G. Under the above assumptions, the induced coalition function coincides with the classical one:

$$V_G^{\mathrm{sem}}(C) = v(C), \qquad \forall C \subseteq N. \tag{5}$$

Consequently, for every agent $i \in N$, the semantic Shapley value satisfies

$$\mathrm{SSV}_i(G) = \phi_i(v). \tag{6}$$

The proof is given in Appendix H.

3.3 A Trace Example with Absorption

We illustrate the role of Boolean absorption with a simple five-agent trajectory. Agent A produces a background-fact node s_1, agent B produces a key-evidence node s_2, and agent C rewrites s_2 into a

prompt node s_3. Agent D then generates an expanded-summary node s_4 from $\{s_1, s_2\}$, while agent E produces the rubric output node s_5 from s_2. The induced semantic generation hypergraph contains

$$\emptyset \xrightarrow{A} s_1, \quad \emptyset \xrightarrow{B} s_2, \quad \{s_2\} \xrightarrow{C} s_3, \quad \{s_1, s_2\} \xrightarrow{D} s_4, \quad \{s_2\} \xrightarrow{E} s_5.$$

Suppose s_5 is the only activated output node. Link-level recovery may produce two candidate agent supports, $\{B, E\}$ and $\{B, C, E\}$. The second support passes through the rewritten node s_3, but it is logically redundant because s_5 is already supported by B and E. Boolean absorption therefore removes the redundant expansion path and leaves

$$f_{s_5}(x) = x_B x_E, \qquad v^{\text{sem}}_{s_5}(x) = \omega(s_5) x_B x_E.$$

Thus, the output-node contribution is split only between the indispensable agents B and E:

$$\text{SSV}_B(G) = \text{SSV}_E(G) = \frac{\omega(s_5)}{2}, \qquad \text{SSV}_A(G) = \text{SSV}_C(G) = \text{SSV}_D(G) = 0.$$

This example shows how SLIC avoids assigning credit to a semantically redundant rewrite. More examples are provided in Appendices J, I and K.

4 Experiments

4.1 Experiment 1: Consistency with Classical Shapley in Medical Multi-Agent Collaboration

Experiment 1 tests the reduction claim in Theorem 1. We construct a controlled medical multi-agent workflow in which the theorem's assumptions are satisfied by construction: agents solve independent semantic subproblems and their outputs are aggregated without interaction effects. We then ask whether SLIC, using only the realized semantic structure, agrees with the classical Shapley attribution for the same workflow.

4.1.1 Task Setup

We use HealthBench (Arora et al., 2025) as the test domain and implement a three-agent workflow in which agents solve separate semantic subproblems independently and their outputs are aggregated afterward. Let $v(S)$ denote the rubric score obtained by an agent subset $S \subseteq N$. Our target quantity is the classical Shapley value

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \big(v(S \cup \{i\}) - v(S)\big). \tag{7}$$

4.1.2 Evaluation Protocol

We evaluate SLIC on 50 HealthBench medical QA cases (`openai/healthbench`), using exact Shapley values (MC-GT) as the reference and comparing against LOO and a holistic LLM judge. We report $L1$ attribution error, Kendall τ_b, and Micro Hit Rate for local logic recovery. Rubric items on which the baseline judge disagrees across repeated runs are excluded from the main analysis (Appendix L.3). Dataset selection, implementation, and model configurations are detailed in Appendix O.

4.1.3 Main Results

Table 1 reports the main results on 50 complex cases. SLIC achieves the best attribution accuracy and ranking consistency among the low-cost methods while retaining the same low-call regime as the holistic judge. Under the same low-call regime as the holistic judge, SLIC reduces the L1 error from 12.90 to 6.61 and raises Kendall τ_b to 0.814, outperforming both LOO and the holistic baseline. At the micro level, SLIC achieves a Micro Hit Rate of 0.857 (150/175), indicating that the aggregate attribution gains are tied to more reliable local support recovery.

We do not scale this consistency experiment further in the main text because MC-style baselines require repeated subset scoring, and their cost grows exponentially with the number of agents. As a supplement, Appendix L.4 reports a 4-agent extension: under corrected-cost accounting, SLIC reduces cost by 93.3% relative to MC while still maintaining low attribution error (L1 = 5.700).

Table 1: Performance comparison on HealthBench. R denotes a normalized per-run cost unit.

Method	Kendall $\tau_b \uparrow$	L1 Error $\downarrow$	API Calls (per case) $\downarrow$	Cost Reduction $\uparrow$
Monte Carlo (GT)	1.000	0.00	$7R$ (avg 95.9)	Baseline
Leave-One-Out (LOO)	0.575	15.08	$4R$ (avg 54.8)	42.9%
Holistic LLM Judge	0.479	12.90	R (avg 13.7)	85.7%
SLIC (Ours)	**0.814**	**6.61**	R (avg 13.7)	**85.7%**

Table 2: Shape alignment (Spearman ρ) between SSV and perturbation-induced score-drop profiles in Experiment 2.

Setting	Weak	Medium	Strong	LOO / C3-style
Meeting decision summarization	0.718	1.000	1.000	0.975
Table-text numerical reasoning	0.667	0.900	0.900	0.900
Rule application	0.500	0.700	0.700	0.700
HealthBench-style medical safety	0.564	0.872	0.872	0.667
Mean	0.612	0.868	0.868	0.810

4.1.4 Error Analysis

Manual inspection of the high-error cases reveals four recurring failure modes: OR-miss under parallel triggering paths, polarity inversion for negative rubric items, asymmetric penalties when multiple agents jointly fail a constraint, and residual judge noise caused by rubric ambiguity. Representative cases and run-level analyses are provided in Appendix L.2–L.3. Overall, the current bottleneck lies more in the induction of logic and the judge instability than in the SSV definition itself.

4.2 Experiment 2: Diagnosing Complex Multi-Role Multi-Agent Architectures

Experiment 2 studies SSV in settings where agent-level contribution is latent rather than directly identifiable. In complex multi-role workflows, the final system score does not come with a ground-truth decomposition into per-agent contributions, and no standard benchmark exists for node importance. We therefore use intervention-induced degradation as an external behavioral probe rather than as a ground-truth attribution target. These probes are imperfect: removal can be confounded by redundancy, and injected corruption can reflect both semantic disruption and structural sensitivity. Nevertheless, if SSV captures functionally consequential node importance, its profile should align with the degradation patterns induced by sufficiently strong interventions.

4.2.1 Task Setup

We construct a diagnostic benchmark with four multi-agent workflows—meeting summarization, numerical reasoning, rule-based judgment, and medical safety QA—with 50 cases each. The settings draw on QMSum (Zhong et al., 2021), TAT-QA (Zhu et al., 2021), LegalBench (Guha et al., 2023), and HealthBench (Arora et al., 2025), respectively. Workflow structures, agent roles, and implementation and model details are provided in Appendix O.

4.2.2 Evaluation Protocol

For each case, we first compute the SSV profile on the unperturbed trajectory. We then intervene on each agent using weak, medium, and strong injected hallucinations, together with an LOO / C3-style absence intervention. Each intervention produces an agent-indexed score-drop profile. We normalize profiles within each setting and report Spearman ρ between SSV and the perturbation-induced profile.

4.2.3 Main Results

Table 2 reports the alignment between the clean-workflow SSV profile and the perturbation-induced score-drop profiles across four task settings and four intervention types. SSV is evaluated through its ability to predict which agents are behaviorally consequential in a realized multi-agent workflow.



Figure 3: Radar plots compare the normalized SSV profile with score-drop profiles induced by weak, medium, strong, and leave-one-out interventions across four evaluation settings. Greater overlap indicates stronger alignment between SSV and intervention-induced degradation patterns.

Across settings, SSV aligns strongly with the profiles induced by medium and strong hallucination interventions (mean $\rho = 0.868$ for both) and also shows substantial alignment with LOO / C3-style absence interventions (mean $\rho = 0.810$). Weak perturbations yield lower and less stable alignment (mean $\rho = 0.612$).

This pattern is consistent with the intended interpretation of SSV. When an intervention sufficiently disrupts an agent's semantic function, the resulting degradation profile closely follows the importance pattern predicted by SSV. Weak perturbations often leave the agent's effective role only partially changed, while absence-based interventions can still be affected by redundancy when other agents compensate for the missing input. Overall, the results indicate that SSV captures functionally meaningful node importance beyond the reduction setting and serves as a practical diagnostic signal for complex multi-role multi-agent workflows.

4.2.4 Structural Analysis

Fig. 3 shows the structural picture behind these numbers. In most healthy workflows, medium and strong perturbation profiles align with SSV, and LOO / C3-style interventions often follow the same trend. The main deviations reveal two distinct structural effects. First, privilege–capability mismatch occurs when an agent performs limited explicit semantic work but still has enough structural control to collapse the system when attacked. This indicates that semantic contribution and structural privilege can diverge as detailed in Appendix M.3.1. Second, removal baselines can fail under redundancy: if two agents provide equivalent critical inputs, removing either one can leave the other sufficient to reconstruct the result, causing LOO / C3-style methods to underestimate both. Additional alignment figures, counterexamples, and raw profile values are provided in Appendix M.2–N and Appendix L.1.

5 Conclusion

This paper studies contribution attribution in LLM-based multi-agent systems and proposes an interpretable semantic allocation framework. Our central observation is that an agent's contribution should not be determined only by its presence in a counterfactual coalition, but by the value-relevant semantics it generates, preserves, or revises along a realized collaborative trajectory. To capture this structure, we formulate attribution on the semantic generation hypergraph induced by the language flow, where redundancy, substitution, joint support, and semantic absorption among agents can be explicitly represented. On the theoretical side, we provide an axiomatic characterization of semantic contribution attribution and show that the resulting allocation is a well-defined Shapley value over semantic support structures. This extends the classical Shapley value from coalition-level utility functions to semantic structures induced by LLM workflows, while recovering the classical Shapley value under standard set-based conditions. Empirically, our results show that the proposed method produces stable and interpretable agent-level contribution signals while substantially reducing the need for costly counterfactual reruns. Overall, this work offers a principled and practical foundation for low-cost contribution analysis in LLM-based multi-agent collaboration.

REFERENCES

Adrian K. Agogino and Kagan Tumer. Unifying temporal and structural credit assignment problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 980–987, 2004.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Robert J. Aumann and Jacques H. Drèze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3(4):217–237, 1974. doi: 10.1007/BF01766876.

Robert J. Aumann and Lloyd S. Shapley. *Values of Non-Atomic Games*. Princeton University Press, 1974.

John F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19 (2):317–343, 1965.

Chi-Min Chan, Jianxuan Yu, Weize Chen, Chunyang Jiang, Xinyu Liu, Weijie Shi, Zhiyuan Liu, Wei Xue, and Yike Guo. Agentmonitor: A plug-and-play framework for predictive and secure multi-agent systems. *arXiv preprint arXiv:2408.14972*, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint arXiv:2308.10848*, 2023.

Yanjun Chen, Yirong Sun, Hanlin Wang, Xinming Zhang, Xiaoyu Shen, Wenjie Li, and Wei Zhang. Contextual counterfactual credit assignment for multi-agent reinforcement learning in llm collaboration. *arXiv preprint arXiv:2603.06859*, 2026.

Pradeep Dubey and Lloyd S. Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, 1979. doi: 10.1287/moor.4.2.99.

Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus, and Michael Wooldridge. Cooperative boolean games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1015–1022, 2008.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, 2018.

Milan Gritta, Debjit Paul, Xiaoguang Li, Lifeng Shang, Jun Wang, and Gerasimos Lampouras. Process evaluation for agentic systems. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026.

Neel Guha et al. LegalBench: A collaboratively built benchmark for measuring legal reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2023.

Paul Harrenstein, Wiebe van der Hoek, John-Jules Ch. Meyer, and Cees Witteveen. Boolean games. In J. van Benthem (ed.), *Proceedings of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pp. 287–298, 2001.

John C. Harsanyi. A bargaining model for cooperative n-person games. In R. Duncan Luce and Albert W. Tucker (eds.), *Contributions to the Theory of Games IV*, volume 40 of *Annals of Mathematics Studies*, pp. 325–356. Princeton University Press, 1959.

John C. Harsanyi. A simplified bargaining model for the n-person cooperative game. *International Economic Review*, 4(2):194–220, 1963. doi: 10.2307/2525487.

Sergiu Hart and Andreu Mas-Colell. Potential, value, and consistency. *Econometrica*, 57(3):589–614, 1989. doi: 10.2307/1911054.

Sirui Hong et al. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Ehud Kalai and Dov Samet. On weighted shapley values. *International Journal of Game Theory*, 16 (3):205–222, 1987. doi: 10.1007/BF01756292.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. *arXiv preprint arXiv:2303.17760*, 2023a.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, December 2023b.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, 2017.

Roger B. Myerson. Graphs and cooperation in games. *Mathematics of Operations Research*, 2(3): 225–229, 1977. doi: 10.1287/moor.2.3.225.

Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Credit assignment for collective multiagent rl with global rewards. In *Advances in Neural Information Processing Systems*, 2018.

Guillermo Owen. Multilinear extensions of games. *Management Science*, 18(5-part-2):64–79, 1972. doi: 10.1287/mnsc.18.5.64.

Guillermo Owen. Values of games with a priori unions. In Rudolf Henn and Otto Moeschlin (eds.), *Essays in Mathematical Economics and Game Theory*, pp. 76–88. Springer, 1977. doi: 10.1007/978-3-642-45494-3_7.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024. doi: 10.18653/v1/2024.acl-long.810.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016.

Lloyd S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker (eds.), *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pp. 307–317. Princeton University Press, 1953.

Lloyd S. Shapley and Martin Shubik. A method for evaluating the distribution of power in a committee system. *American Political Science Review*, 48(3):787–792, 1954.

Richard S. Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 1984.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. MedAgents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 599–621, 2024.

David H. Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03):265–279, 2001.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Yingxuan Yang, Bo Huang, Siyuan Qi, Chao Feng, Haoyi Hu, Yuxuan Zhu, Jinbo Hu, Haoran Zhao, Ziyi He, Xiao Liu, et al. Understanding and optimizing agentic workflows via shapley value. *arXiv preprint arXiv:2502.00510*, 2025.

H. Peyton Young. Monotonic solutions of cooperative games. *International Journal of Game Theory*, 14(2):65–72, 1985.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833, 2014.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *International Conference on Learning Representations*, 2025a.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. AFlow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*, 2024.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. In *International Conference on Machine Learning*, pp. 76583–76599. PMLR, 2025b.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of NAACL-HLT*, pp. 5905–5921, 2021.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of ACL-IJCNLP*, pp. 3277–3287, 2021.

AI USE STATEMENT

We used generative AI tools to assist with refining the conceptual framework. The authors independently reviewed the theoretical claims and proofs, checked the experimental design and results, and revised all AI-assisted text. The authors take full responsibility for the content of this paper.

A RELATED WORKS

LLM-based multi-agent systems and agentic workflows. Recent multi-agent LLM frameworks show that capability gains arise not only from scaling the base model, but also from explicit coordination mechanisms. CAMEL, AutoGen, and MetaGPT demonstrate the utility of role specialization, inter-agent conversation, and modular decomposition (Li et al., 2023a; Wu et al., 2023; Hong et al., 2024). Subsequent systems such as ChatDev and AgentVerse treat collaboration itself as the primary computational object and study how structured communication shapes collective behavior (Qian et al., 2024; Chen et al., 2023). Other work investigates specific coordination ingredients, including theory-of-mind reasoning across agents (Li et al., 2023b), communication pruning for reducing coordination overhead (Zhang et al., 2025a), and automated workflow search and construction (Zhang et al., 2024). In domain-specific settings, MedAgents shows that carefully designed multi-expert collaboration can be effective even in high-stakes medical reasoning (Tang et al., 2024). Taken together, these studies establish that the performance of agentic systems depends crucially on workflow structure, communication pathways, and the handling of intermediate artifacts, rather than merely on which agents are present.

Cooperative game theory and structured cooperation. The theoretical basis for our work comes from cooperative game theory, where the Shapley value remains the canonical starting point for contribution allocation (Shapley, 1953). Classical extensions already suggest that structured cooperation requires structured value notions. Harsanyi's bargaining and dividend-based formulations decompose coalition value into finer interaction terms (Harsanyi, 1959; 1963). Aumann and Shapley analyze value in non-atomic settings, and Young gives an axiomatic perspective based on monotonicity (Aumann & Shapley, 1974; Young, 1985). Owen's multilinear extension provides an algebraic representation of games that is particularly relevant when allocations are computed from decomposed interaction terms (Owen, 1972). When cooperation is constrained by predefined groups or communication relations, classical work already departs from the plain characteristic-function view: Aumann and Drèze study coalition structures, Myerson studies graph-restricted cooperation, and Owen extends value allocation to games with a priori unions (Aumann & Drèze, 1974; Myerson, 1977; Owen, 1977). Further generalizations include weighted Shapley values and potential-based characterizations (Kalai & Samet, 1987; Hart & Mas-Colell, 1989). Even in voting and committee settings, the Shapley–Shubik and Banzhaf lines show that influence depends on structural notions of pivotality rather than on presence alone (Shapley & Shubik, 1954; Banzhaf, 1965; Dubey & Shapley, 1979). This broader literature supports our central premise that once cooperation is mediated by structure, the value object itself should be defined on that structure.

Credit assignment from reinforcement learning to LLM collaboration. A second closely related tradition is credit assignment. Early work on temporal credit assignment in reinforcement learning formalized the difficulty of tracing delayed outcomes back to individual decisions (Sutton, 1984). Later work generalized this perspective to structural settings and collective systems, showing that reward redistribution and payoff design must account for interaction topology rather than only individual actions (Agogino & Tumer, 2004; Wolpert & Tumer, 2001). In multi-agent reinforcement learning, COMA uses a counterfactual baseline to isolate the marginal effect of each agent action on joint return (Foerster et al., 2018), and subsequent work studies credit assignment under global rewards and collective objectives (Nguyen et al., 2018). More recently, this counterfactual line has been adapted to language-mediated collaboration: C3 estimates message-level responsibility through fixed-continuation replay and leave-one-out interventions in LLM collaboration (Chen et al., 2026). These approaches provide useful causal intuitions, but they generally identify contribution by rerunning altered systems, which becomes expensive and unstable in large language-based workflows.

B Limitations and Discussion

SCG and SLIC are intended for attribution on realized language-flow traces, not for all forms of multi-agent evaluation. The first limitation is semantic recovery. SLIC assumes that value-bearing semantic nodes and their support relations can be extracted with sufficient reliability. This is reasonable for rubric-based tasks with explicit intermediate outputs, but remains difficult when support is implicit, negated, masked by later text, or distributed across paraphrases. In those cases, errors in the support logic directly affect the SSV.

The second limitation is the value-reading interface. Our framework attributes the value that the interface reads from O. If the rubric is incomplete, ambiguous, or unstable between judge calls, then the resulting attribution inherits that uncertainty. This is not a failure of the Shapley allocation step, but it limits the maximum reliability of any method that depends on the same observed value signal.

Despite these limitations, the main message is that contribution attribution in LLM-based multi-agent systems need not be reduced to black-box counterfactual reruns. By treating language flow as the observable layer and semantic support as the value-bearing structure, SCG gives a concrete object for attribution, and SLIC provides a fast single-trajectory computation path. This speed matters: by exposing which semantic transformations support task value without repeated reruns, SSV makes downstream workflow auditing, robustness analysis, agent pruning, and contribution-aware system redesign practically possible. We leave these intervention-oriented uses to future work.

C Workflow and Language Flow

This appendix fills in the workflow, language flow, and value-node definitions that are compressed in the main text.

For any execution position u, let the input and output satisfy

$$x_u,\ y_u \in \mathcal{X}, \qquad y_u = F^a(x_u; \theta^a).$$

Here x_u and y_u denote the semantic states before and after this step, a is the acting agent, and θ^a denotes the stable configuration associated with that agent.

The language flow in the main text is written as

$$\mathcal{T} = (L, \mathcal{H}),$$

where $L \subseteq \mathcal{X}$ is the set of semantic states propagated along the workflow channel, and $\mathcal{H} = \{h_u : u \in V\}$ is the set of corresponding agent actions. Workflow describes the collaboration structure among agents, whereas language flow describes how those agent actions act on semantic states.

We then introduce a task value function on the semantic state space:

$$v : \mathcal{X} \to \mathbb{R}.$$

We further assume that v admits a finite discriminative decomposition, namely that there exist finitely many discriminators

$$r_k : \mathcal{X} \to \{0, 1\}, \qquad k = 1, \dots, m,$$

and weights $w_k \in \mathbb{R}$ such that

$$v(z) = \sum_{k=1}^{m} w_k\, r_k(z), \qquad \forall z \in \mathcal{X}.$$

In practice, $\{r_k\}_{k=1}^m$ can be viewed as a set of rubric-style yes-or-no evaluation items.

For any output semantic state y_u, let

$$O_u$$

denote the set of local semantic components in y_u that can activate an evaluation item. We call the elements of O_u value nodes. Let

$$O := \bigcup_u O_u$$

be the set of all value nodes.

D Recursive Closure Construction from Language Flow

Next, we do not keep all semantics, but only those lying on the generation chains leading to the value nodes. Assumption 1 in the main text can be restated here as follows: for any execution position u, the local semantic components in the output state y_u that enter the task value formation process can be identified and separated; each such local semantic component $s \subseteq y_u$ belongs to one of three categories, namely native, derived, or persistent semantics. If s is derived or persistent, its previous semantic source in the input state x_u can be identified.

Accordingly, for any $h_u \in \mathcal{H}$, define the source mapping

$$\mathrm{src}_{h_u}$$

as follows. For any local semantic component $s \subseteq y_u$, let

$$\mathrm{src}_{h_u}(s) = \begin{cases} \varnothing, & s \text{ is native}, \\ \{\tilde{s}\}, & s \text{ is persistent}, \\ P,\ P \subseteq x_u, & s \text{ is derived}, \end{cases}$$

where $\tilde{s}$ denotes the same old semantic unit preserved from the input state x_u, and P denotes the set of old semantic sources that generate s.

Starting from the value nodes, define recursively

$$S^{(0)} := O,$$

$$S^{(t+1)} = S^{(t)} \cup \bigcup_{u} \bigcup_{\substack{s \in S^{(t)} \\ s \subseteq y_u}} \mathrm{src}_{h_u}(s), \qquad t = 0, 1, 2, \ldots,$$

and let

$$S := \bigcup_{t \geq 0} S^{(t)}.$$

In other words, S is the minimal semantic set obtained by backward recursive closure from the value-node set O along language flow.

After obtaining the node set S, we further extract effective input–output relations from local operations in language flow. Specifically, for any $h_u \in \mathcal{H}$, if there exist

$$Q \subseteq S, \qquad R \subseteq S,$$

where Q comes from the input state x_u, R comes from the output state y_u, and the nodes in R are supported by Q at the current step, then

$$\ell = (Q, R)$$

is called a semantic link induced by h_u. Let $\mathcal{L}$ be the set of all such links.

This yields the semantic generation graph

$$G = (S, \mathcal{L}, N; \alpha, \omega, O, I),$$

where $\alpha : \mathcal{L} \to N$ is the link ownership map, ω is the node-weight function, and I is the set of initial semantic nodes. In the main text, for notational consistency, we simply write $\mathcal{L}$ as L.

E Counterfactual Validation of Semantic Support Relations

The semantic analyzer recovers fine-grained semantic points and directed support links from a realized multi-agent trajectory. Because these links are identified from language rather than observed interventions, we additionally test whether they exhibit the counterfactual behavior implied by their labels. This experiment evaluates graph reconstruction before link-to-agent projection and Semantic Shapley computation.

Evaluation set. We reuse 160 realized workflows from four benchmark-style task families: HealthBench-style medical safety, LegalBench-style rule application, QMSum-style meeting synthesis, and TAT-QA-style numerical reasoning, with 40 cases per family. For the support-removal test, we select one analyzer-recovered relation per case using only graph structure and target grounding, without access to replay outcomes. This produces 120 transformed and 40 preserved relations. Of these 160 relations, 118 have multiple input semantic points (maximum arity 11), so the evaluation covers semantic hyperedges rather than only pairwise links. We separately evaluate one new event per case.

Counterfactual replay. Let $\ell = (U \to v)$ denote a recovered support link from a set of predecessor semantics U to a target semantic point v. We replay the same target agent under three conditions while holding its role prompt, task input, generation model, and all non-target context fixed:

1. **Factual** (C_{F}): the original predecessor context;
2. **Support removal** (C_{-U}): every grounded carrier of U, including aliases and repeated mentions, is removed; and
3. **Sham** (C_{S}): the context passes through the same editing procedure, but its semantic propositions are preserved.

The sham arm controls for output changes caused by rewriting the context rather than removing the proposed support set. Every edited context must pass an separate audit: all support semantics are absent, the target is not injected, at least 80% of unrelated propositions are retained, the sham preserves the original propositions, and no fragmentary sentence shell remains.

For each arm $a \in \{\mathrm{F}, -U, \mathrm{S}\}$, we generate $K = 3$ outputs and use semantic presence, rather than exact string matching, to estimate

$$\widehat{p}_a(v) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}\Big[v \text{ is expressed in } y_a^{(k)}\Big]. \tag{8}$$

We define the edit-controlled effect as

$$\widehat{\Delta}_{\mathrm{spec}}(U \to v) = \widehat{p}_{\mathrm{S}}(v) - \widehat{p}_{-U}(v). \tag{9}$$

A preserved or transformed relation is confirmed when $\widehat{p}_{\mathrm{F}}(v) \geq 2/3$, $\widehat{p}_{\mathrm{F}}(v) - \widehat{p}_{-U}(v) \geq 1/3$, and $\widehat{\Delta}_{\mathrm{spec}}(U \to v) \geq 1/3$.

The two relation types use the same intervention but differ semantically. Preserved means that at least one predecessor already expresses the target proposition, possibly as a paraphrase. Transformed means that the target differs from its inputs and is obtained by deriving, combining, classifying, computing, specializing, or operationalizing them. We therefore also reclassify each input-output pair and report type agreement separately from counterfactual support confirmation.

Testing new events. A new point has no predecessor semantic support by definition. Deleting the entire task would therefore be an invalid test. Instead, the anchored-null replay retains the exogenous task input, task identity, domain, agent role, and format constraints, while removing all semantic payloads produced by predecessor agents. A new event is confirmed when the target appears in both factual and anchored-null replay with probability at least $2/3$. No independently sampled sham arm is required for this test because no semantic editing operation is applied.

All replay generation, semantic editing, and semantic-presence judgments use Gemini 2.5 Flash at temperature zero. Each factual, removal, sham, or anchored-null condition is sampled three times. The trajectories and recovered graphs are fixed before replay. All valid but unconfirmed relations remain in the corresponding denominators, while technical failures are reported through the planned and valid counts.

Results. The support-removal test confirms 149 of 160 recovered links (93.1%, 95% bootstrap CI $[88.8\%, 96.9\%]$). The target semantic appears with mean probability .958 in factual replay and .960 in sham replay, but only .021 after removing the recovered support set. Thus, the decrease is specific to semantic support removal rather than context rewriting. Relation type agrees with the separate

Table 3: Counterfactual validation by semantic relation. Confirmation-rate intervals are 95% bootstrap confidence intervals over relation events. For new relations, $\widehat{p}_{-U}$ denotes the anchored-null replay and the sham arm is not applicable.

Relation	Planned	Valid	Confirmed	Rate [95% CI]	$\widehat{p}_{\mathrm{F}}$	$\widehat{p}_{-U}$	$\widehat{p}_{\mathrm{S}}$	Type match
Transformed	120	120	110	.917 [.867, .958]	.950	.028	.956	118/120
Preserved	40	40	39	.975 [.925, 1.000]	.983	.000	.975	40/40
New	160	159	154	.969 [.937, .994]	.969	.969	–	159/159

Table 4: Validation across task families. The support-removal columns combine transformed and preserved relations. "New conf." reports confirmation under anchored-null replay.

Task family	N_{sup}	Support conf.	$\widehat{p}_{\mathrm{F}}$	$\widehat{p}_{-U}$	$\widehat{p}_{\mathrm{S}}$	N_{new}	New conf.
HealthBench-style	40	36/40	.917	.008	.925	39	39/39
LegalBench-style	40	37/40	.958	.025	.967	40	37/40
QMSum-style	40	38/40	.958	.000	.950	40	38/40
TAT-QA-style	40	38/40	1.000	.050	1.000	40	40/40
Overall	160	149/160	.958	.021	.960	159	154/159

semantic reclassification for 158 of 160 events; requiring both counterfactual confirmation and type agreement retains 147 of 160 events (91.9%).

For new relations, 154 of 159 valid events (96.9%) remain present after all predecessor-agent semantics are removed. All five valid misses fail to reproduce the target in both the factual and anchored-null arms, rather than exhibiting a reversal in which predecessor removal alone suppresses the target. One planned HealthBench-style event is recorded as a technical failure because the model service blocked the request; it is not counted as a valid replay. Across task families, support confirmation ranges from 90.0% to 95.0%, and new-relation confirmation ranges from 92.5% to 100%.

Interpretation and scope. These results provide a behavioral reliability check for the semantic hypergraph recovered by the analyzer. In the fixed-agent replay environment, the predecessor sets identified for preserved and transformed links are usually necessary for reproducing their targets, whereas points labeled new usually remain reproducible without predecessor-agent semantics. This is a precision-type validation of proposed relations. It does not establish that an observed trajectory admits a unique causal graph, nor does it measure exhaustive recall over every possible predecessor set. The three-sample estimate also provides a coarse behavioral probability. We therefore use this experiment to support the reliability of recovered links under controlled replay, not to claim complete causal identification of the semantic graph.

F Sensitivity to the Semantic Analyzer

We extend Experiment 2 with an analyzer-model sensitivity study on the same 200 realized multi-agent workflows, comprising 1,950 rubric-level graphs. The trajectories, rubric-judge outputs, manually constructed outputs under hallucination interventions, and counterfactual and rerun score drops are held fixed. Only the LLM used to extract the semantic graph is changed. We use Gemini 2.5 Pro as the reference analyzer and compare it with four replacement analyzers.

The agent-level SSV rankings remain similar to those obtained with Gemini 2.5 Pro: the rank correlation exceeds 0.81 for every replacement analyzer and reaches 0.959 for Gemini 2.5 Flash. The ℓ_1 differences show that the attribution magnitudes can still vary across analyzers. Thus, the evidence supports stability of the relative agent rankings, rather than invariance of the exact allocations.

Agreement after agent projection. We compare overlap between the raw semantic links extracted by two analyzers with overlap after mapping those links to their associated agents. For replacement

Table 5: Sensitivity to the semantic analyzer. Each replacement analyzer is compared with Gemini 2.5 Pro on the same 200 workflows and 1,950 rubric-level graphs. Brackets show the reported uncertainty intervals.

Analyzer	SSV rank ρ	SSV ℓ_1	Raw-link Jaccard	Link–agent Jaccard
Gemini 2.5 Flash-Lite	0.831 [0.814, 0.848]	0.354 [0.331, 0.377]	0.056 [0.049, 0.063]	0.613 [0.591, 0.635]
Gemini 2.5 Flash	0.959 [0.947, 0.969]	0.112 [0.098, 0.126]	0.550 [0.525, 0.575]	0.850 [0.833, 0.867]
Llama 4 Scout	0.816 [0.796, 0.836]	0.389 [0.367, 0.412]	0.060 [0.053, 0.068]	0.610 [0.586, 0.634]
Llama 4 Maverick	0.853 [0.831, 0.873]	0.362 [0.339, 0.385]	0.099 [0.089, 0.110]	0.664 [0.647, 0.681]

analyzer m, let

$$\Delta J^{(m)} = J^{(m)}_{\text{link-agent}} - J^{(m)}_{\text{raw-link}}. \tag{10}$$

In the order shown in Table 5, the average differences are 0.557, 0.300, 0.550, and 0.565, respectively ($p_{\text{Wilcoxon}} < 10^{-33}$). Overlap is therefore higher in the agent-associated representation for all four analyzers.

This distinction matters because analyzers can describe the same local reasoning at different levels of granularity. For example, one analyzer may represent a patient's bleeding risk and the resulting dose adjustment as one derived semantic node, while another separates the risk assessment from the dose adjustment. Their raw links need not coincide, even if the corresponding transformations are assigned to the same agent. Accordingly, we do not assume that the reconstructed semantic graph is invariant across analyzers. The observed robustness concerns the resulting agent-level SSV rankings.

G Full Derivation of Semantic Shapley Value

The main text only gives the final formula for SSV. Here we provide the intermediate derivation.

A. From link supports to agent supports. Fix an output node $o_k \in O$. Let

$$\mathcal{C}_{o_k} \subseteq 2^L \setminus \{\emptyset\}$$

be the set of all minimal link-level conjunctive supports that semantically support o_k. For any $C \in \mathcal{C}_{o_k}$, define the corresponding agent set by

$$N(C) := \{\alpha(\ell) : \ell \in C\} \subseteq N.$$

Collect all such agent sets and remove strict redundant supersets, yielding the minimal subject support family

$$\mathcal{M}_{o_k} := \min_{\subseteq}\{N(C) : C \in \mathcal{C}_{o_k}\}.$$

Here, $\min_{\subseteq}$ means keeping only inclusion-minimal sets. In the main-text example, the absorption of $\{B, E\}$ over $\{B, C, E\}$ is a direct instance of this step.

B. From the support family to support logic. For each output node o_k, define the agent-level support function

$$f_{o_k}(x) = \bigvee_{M \in \mathcal{M}_{o_k}} \bigwedge_{i \in M} x_i, \qquad x \in \{0,1\}^{|N|}.$$

Here $x_i = 1$ means that agent i is included in the current configuration. Therefore,

$$f_{o_k}(x) = 1$$

if and only if the configuration x is semantically sufficient to support the output node o_k. Using the weight $\omega(o_k)$ of the output node, we obtain the corresponding local value function

$$v^{\text{sem}}_{o_k}(x) := \omega(o_k) f_{o_k}(x).$$

C. From coefficient expansion to SSV. For each o_k, the function f_{o_k} has a unique multilinear expansion:

$$f_{o_k}(x) = \sum_{T \subseteq N} c_T^{(o_k)} \prod_{i \in T} x_i,$$

where the coefficients are given by Möbius inversion:

$$c_T^{(o_k)} = \sum_{R \subseteq T} (-1)^{|T|-|R|} f_{o_k}(\mathbf{1}_R).$$

Here $\mathbf{1}_R$ is the indicator vector that is 1 exactly on R. The meaning of this expansion is that each monomial

$$\prod_{i \in T} x_i$$

corresponds to a basic joint support unit. If its coefficient is $c_T^{(o_k)}$, then the total weight of that unit is $\omega(o_k) c_T^{(o_k)}$.

For any joint support unit T, its weight is split evenly among the agents in T. Hence the allocation that agent i receives from output node o_k is

$$\phi_i^{\text{sem}}(o_k; G) = \omega(o_k) \sum_{T \subseteq N:\, i \in T} \frac{c_T^{(o_k)}}{|T|}.$$

Summing over all activated output nodes gives the graph-level semantic Shapley value used in the main text:

$$\text{SSV}_i(G) = \sum_{o_k \in O: x_{o_k}(G)=1} \phi_i^{\text{sem}}(o_k; G) = \sum_{o_k \in O: x_{o_k}(G)=1} \omega(o_k) \sum_{T \subseteq N:\, i \in T} \frac{c_T^{(o_k)}}{|T|}.$$

Therefore, SSV is not obtained by directly averaging over minimal support sets. Instead, it keeps the full Boolean support logic first, and then performs analytical allocation on the multilinear / Möbius basis. This is precisely why it can handle shared predecessors, parallel support paths, and absorption relations.

H Proof of Theorem 1

Proof. Fix an activated output node o_k. Consider its agent-level Boolean support logic

$$f_{o_k}(x) = \sum_{T \subseteq N} c_T^{(o_k)} \prod_{j \in T} x_j, \qquad x \in \{0,1\}^{|N|}.$$

Under the set-based, fully observable, and no-masking / no-order-dependence conditions stated in the theorem, whether the node o_k is hit depends only on the subset of participating agents. Hence f_{o_k} can be naturally viewed as a Boolean characteristic function on the coalition space 2^N: for any $C \subseteq N$, let $\mathbf{1}_C \in \{0,1\}^{|N|}$ be its indicator vector and define the coalition hit function

$$v_{o_k}(C) := f_{o_k}(\mathbf{1}_C).$$

Thus $v_{o_k} : 2^N \to \{0,1\}$ is a standard characteristic function of a cooperative game.

From the multilinear expansion of f_{o_k}, for any $C \subseteq N$,

$$v_{o_k}(C) = f_{o_k}(\mathbf{1}_C) = \sum_{T \subseteq N} c_T^{(o_k)} \prod_{j \in T} \mathbf{1}_C(j).$$

Note that

$$\prod_{j \in T} \mathbf{1}_C(j) = \begin{cases} 1, & T \subseteq C, \\ 0, & T \not\subseteq C, \end{cases}$$

and therefore

$$v_{o_k}(C) = \sum_{T \subseteq C} c_T^{(o_k)}.$$

Now for any $C \subseteq N \setminus \{i\}$,

$$v_{o_k}(C \cup \{i\}) - v_{o_k}(C) = \sum_{T \subseteq C \cup \{i\}} c_T^{(o_k)} - \sum_{T \subseteq C} c_T^{(o_k)}.$$

All terms not containing i cancel, hence

$$v_{o_k}(C \cup \{i\}) - v_{o_k}(C) = \sum_{T \subseteq C \cup \{i\}:\ i \in T} c_T^{(o_k)}.$$

Substitute this into the classical Shapley marginal formula. Let $n = |N|$. Then

$$\phi_i(v_{o_k}) = \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!(n - |C| - 1)!}{n!} \big(v_{o_k}(C \cup \{i\}) - v_{o_k}(C)\big)$$

$$= \sum_{C \subseteq N \setminus \{i\}} \frac{|C|!(n - |C| - 1)!}{n!} \sum_{T \subseteq C \cup \{i\}:\ i \in T} c_T^{(o_k)}.$$

Swapping the order of summation gives

$$\phi_i(v_{o_k}) = \sum_{T \ni i} c_T^{(o_k)} \sum_{C:\, T \setminus \{i\} \subseteq C \subseteq N \setminus \{i\}} \frac{|C|!(n - |C| - 1)!}{n!}.$$

It remains to compute the inner coefficient. Let $|T| = t$. If $T \setminus \{i\} \subseteq C \subseteq N \setminus \{i\}$ and $|C| = k$, then k ranges from $t - 1$ to $n - 1$, and the number of such sets C is

$$\binom{n - t}{k - t + 1}.$$

Hence the inner sum is

$$\sum_{k=t-1}^{n-1} \binom{n - t}{k - t + 1} \frac{k!(n - k - 1)!}{n!}.$$

Let $r = k - t + 1$, so $r = 0, \ldots, n - t$. Then the above becomes

$$\frac{(n - t)!}{n!} \sum_{r=0}^{n-t} \frac{(r + t - 1)!}{r!}.$$

Using

$$\frac{(r + t - 1)!}{r!} = (t - 1)! \binom{r + t - 1}{t - 1}$$

and the hockey-stick identity

$$\sum_{r=0}^{M} \binom{r + t - 1}{t - 1} = \binom{M + t}{t},$$

with $M = n - t$, we obtain

$$\frac{(n - t)!}{n!} \sum_{r=0}^{n-t} \frac{(r + t - 1)!}{r!} = \frac{(n - t)!(t - 1)!}{n!} \binom{n}{t} = \frac{1}{t} = \frac{1}{|T|}.$$

Therefore,

$$\phi_i(v_{o_k}) = \sum_{T \ni i} \frac{c_T^{(o_k)}}{|T|}.$$

Multiplying back the node weight $\omega(o_k)$ yields

$$\phi_i^{\text{sem}}(o_k; G) = \omega(o_k) \phi_i(v_{o_k}) = \omega(o_k) \sum_{T \ni i} \frac{c_T^{(o_k)}}{|T|},$$

which is exactly the node-level semantic allocation given in the main text and in Appendix G.

Finally, aggregate over all activated output nodes. Let

$$v(C) := \sum_{o_k \in O:\, x_{o_k}(G)=1} \omega(o_k)\, v_{o_k}(C), \qquad C \subseteq N.$$

By additivity of the classical Shapley value,

$$\phi_i(v) = \sum_{o_k \in O:\, x_{o_k}(G)=1} \omega(o_k)\phi_i(v_{o_k}) = \sum_{o_k \in O:\, x_{o_k}(G)=1} \phi_i^{\mathrm{sem}}(o_k; G) = \mathrm{SSV}_i(G).$$

Therefore, under the reduction conditions, the SSV equals the Shapley value of the corresponding classical coalition game. □

I Supplementary Structural Examples

I.1 A Linear Example of Backward Extraction of Semantic Node Types

This example corresponds to the simplest single-output chain structure in the thesis. Fig. 4 places the workflow, the language flow, and the semantic generation hypergraph side by side, making it easy to see that the output node is recovered through a single main chain. As a result, the predecessor closure, the initial node set I, and the final hyperedge set L are all straightforward.

I.2 A Parallel Multi-Value-Node Example

Fig. 5 shows a multi-output parallel structure. Different branches produce their own semantic nodes, but only those that enter the value-formation chain are retained in S. It also shows that an agent may still have no substantive contribution in the final hypergraph even if it "produces content," as long as that content does not form an effective semantic role for any value-relevant node.

J Supplementary Logic Examples

J.1 Link-Level Support

Consider a collaborative decision task whose goal is to determine whether a technical plan is feasible. Fig. 6 illustrates this example.Let the semantic nodes be

$$\begin{aligned} s_0 &= \text{task background and constraints are given,} \\ s_1 &= \text{the core module can be completed on time,} \\ s_2 &= \text{the budget constraint can be satisfied,} \\ s_3 &= \text{the existing system can directly host the plan,} \\ s_4 &= \text{the plan is feasible.} \end{aligned}$$

Assume that the initial node set contains at least s_0 and s_3. The four semantic links in the graph are

$$\ell_1 : \{s_0\} \to \{s_1\}, \qquad \ell_2 : \{s_0\} \to \{s_2\},$$

$$\ell_3 : \{s_1, s_2\} \to \{s_4\}, \qquad \ell_4 : \{s_3\} \to \{s_4\}.$$

Hence

$$\{\ell_1, \ell_2, \ell_3\} \Rightarrow s_4, \qquad \{\ell_4\} \Rightarrow s_4,$$

and therefore

$$\mathcal{C}_{s_4} = \big\{\{\ell_1, \ell_2, \ell_3\}, \{\ell_4\}\big\}, \qquad s_4 \iff (\ell_1 \wedge \ell_2 \wedge \ell_3) \vee \ell_4.$$

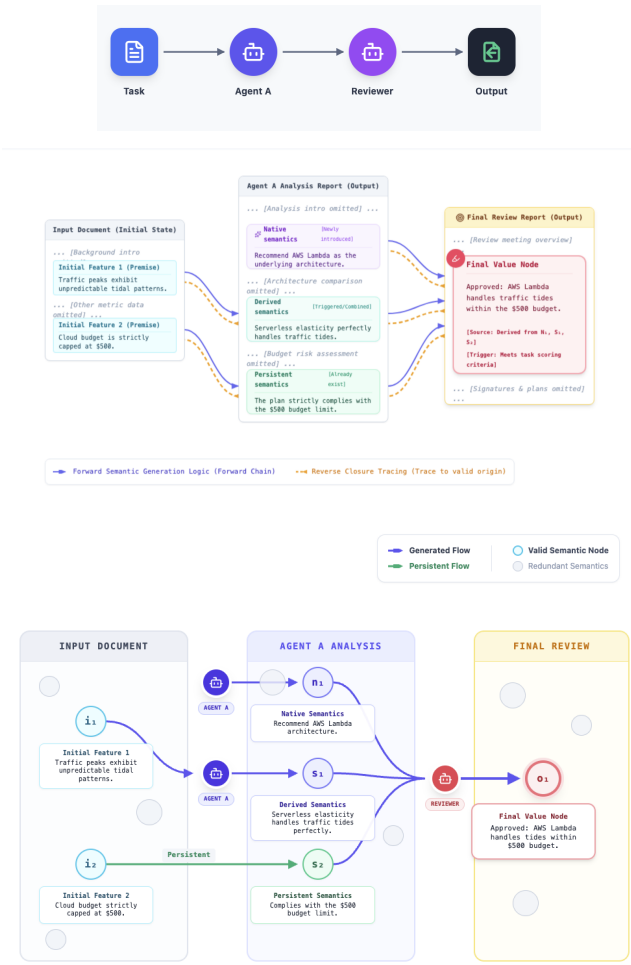


Figure 4: A linear workflow, its dual language flow, and the induced semantic generation hypergraph.

J.2 Absorption from Link Level to Agent Level

Continue with the above example. If the ownership of the four semantic links satisfies

$$\alpha(\ell_1) = A, \qquad \alpha(\ell_2) = B, \qquad \alpha(\ell_3) = C, \qquad \alpha(\ell_4) = A,$$

then

$$N(\{\ell_1, \ell_2, \ell_3\}) = \{A, B, C\}, \qquad N(\{\ell_4\}) = \{A\}.$$

Hence the support logic of node s_4 at the agent level is

$$s_4 \Longleftrightarrow (A \wedge B \wedge C) \vee A.$$

By Boolean absorption,

$$(A \wedge B \wedge C) \vee A = A,$$

and thus the final minimal support term is

$$\mathcal{M}_{s_4} = \{\{A\}\}, \qquad f_{s_4}(x) = x_A.$$

Fig. 7 illustrates this example

K Supplementary Medical Case

This section gives a more complete medical multi-agent example to illustrate how SLIC constructs a semantic generation hypergraph from a single realized collaborative trajectory and then computes the corresponding SSV. Fig. 8 shows this workflow.

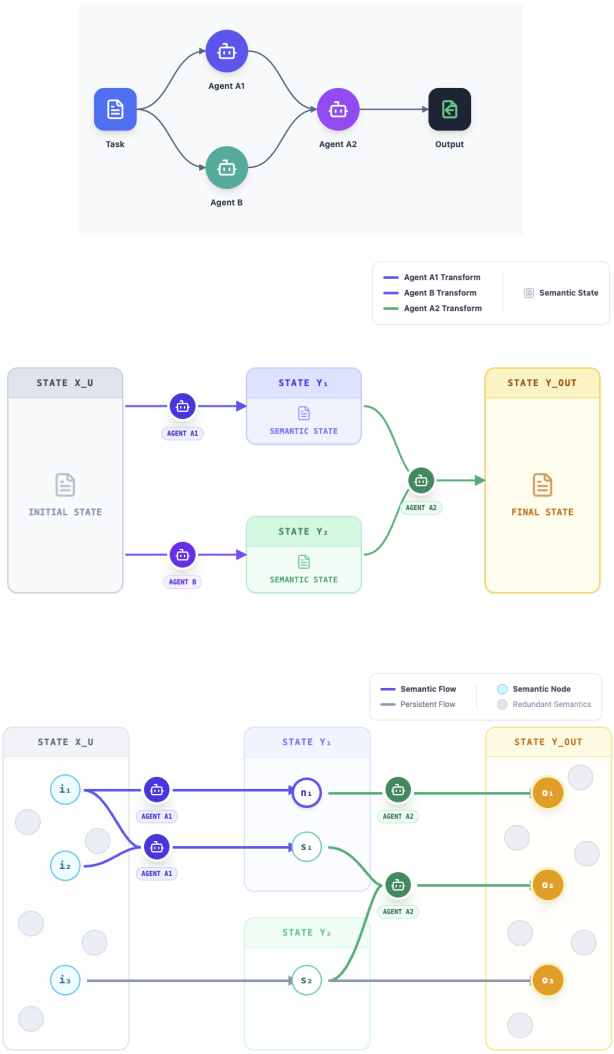


Figure 5: A parallel workflow, its dual language flow, and the induced semantic generation hypergraph with multiple value nodes.

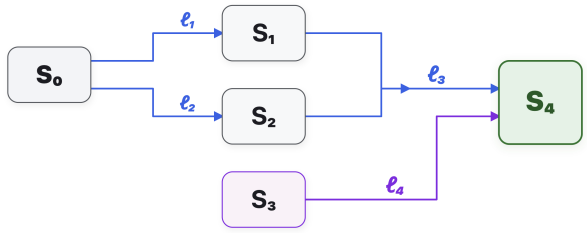


Figure 6: A supplementary example of link-level support.

Node Extraction and Transformation in the Semantic Network

During actual multi-agent collaboration, we introduce an independent large language model as a side-channel parser. Its task is to filter irrelevant analytical traces from the multi-turn dialogue and

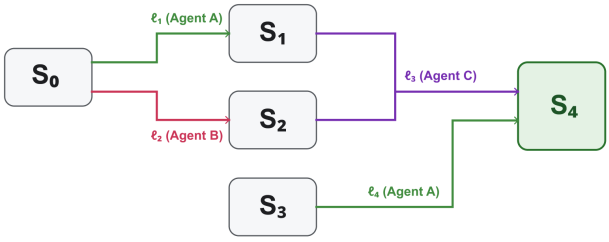


Figure 7: A supplementary example of absorption from link-level logic to agent-level logic.

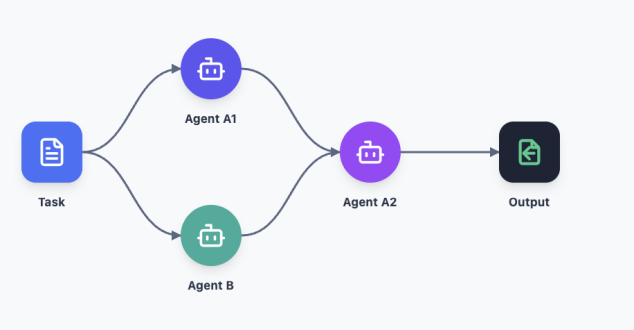


Figure 8: Workflow of the medical multi-agent collaboration.

extract the structured semantic state set that truly drives task progress:

$$S = \{s_0, s_1, s_2, s_3, s_4, s_5\}.$$

- s_0: **Initial input**, namely the original medical dialogue record, with $\omega(s_0) = 0$.
- s_1: **Intermediate semantics**, representing **[patient information]**, e.g., advanced age (72), low body weight (50kg), and hypertension (150/90), with $\omega(s_1) = 0$.
- s_2: **Intermediate semantics**, representing **[symptoms]**, e.g., typical constrictive chest pain, cold sweat, and a sense of impending death, with $\omega(s_2) = 0$.
- s_3: **Valuable output node** O_1, representing **[diagnosis]**, e.g., acute myocardial infarction (AMI), with weight $\omega(s_3) = 10$.
- s_4: **Intermediate semantics**, representing **[medication dosage]**, e.g., the anticoagulant dose should be reduced by 50% to avoid bleeding risk, with $\omega(s_4) = 0$.
- s_5: **Valuable output node** O_2, representing **[prescription]**, e.g., the final prescription includes 300mg aspirin and reduced-dose enoxaparin, with weight $\omega(s_5) = 20$.

A text-to-graph transformation mapping Φ. The parser does more than extract isolated propositional nodes. More importantly, it uses causal connectives and contextual logic in natural language to identify semantic links $\mathcal{L}$ between nodes, that is, how semantics continue to evolve along the trajectory. Consider the output of Agent B:

> *"Based on the current symptoms, the preliminary diagnosis is acute myocardial infarction. Meanwhile, the patient's weight is only 50kg, so the anticoagulant dosage should be reduced by 50% to avoid bleeding risk."*

Scanning this text, the mapping Φ automatically performs the following graph-structured transformation:

1. **Extract new states:** instantiate node s_3 (AMI diagnosis) and s_4 (dose reduction by half).

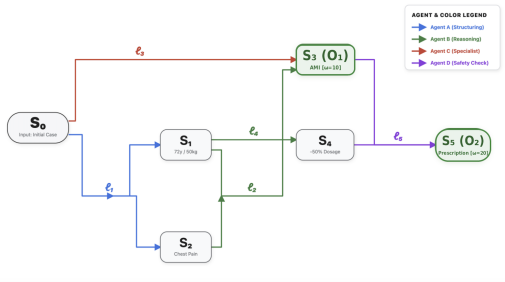


Figure 9: The semantic graph for the medical case.

2. **Identify link ℓ_2:** from the trigger phrase "based on the current symptoms," take the existing symptom node s_2 together with the related sign node s_1 as input and point to output s_3, thus generating the directed link $\ell_2 = (\{s_1, s_2\}, \{s_3\})$.
3. **Identify link ℓ_4:** from the factual dependency "weight is only 50kg," take the sign node s_1 as input and point to output s_4, thus generating the link $\ell_4 = (\{s_1\}, \{s_4\})$.

In other words, generating semantic links is itself part of an agent's behavior. After traversing the global log, the semantic transformation events that actually occurred form the following graph structure:

- $\ell_1 = (\{s_0\}, \{s_1, s_2\})$: decompose the original text into signs and symptoms.
- $\ell_2 = (\{s_1, s_2\}, \{s_3\})$: infer a diagnosis from signs and symptoms.
- $\ell_3 = (\{s_0\}, \{s_3\})$: jump directly from the original text to the diagnosis.
- $\ell_4 = (\{s_1\}, \{s_4\})$: derive a dosage constraint from a sign.
- $\ell_5 = (\{s_3, s_4\}, \{s_5\})$: integrate diagnosis and dosage constraint into the final prescription.

This semantic generation hypergraph is illustrated in Fig. 9.

From Link Paths to Agent-Level Logic

Under the SCG framework, an agent is not identical to a node or an edge; rather, it is the executor of a connection. Through the ownership mapping

$$\alpha : \mathcal{L} \to N,$$

the five semantic links above are assigned as

$$\alpha(\ell_1) = A, \qquad \alpha(\ell_2) = B, \qquad \alpha(\ell_3) = C, \qquad \alpha(\ell_4) = B, \qquad \alpha(\ell_5) = D.$$

Based on this mapping, we first write the link-level logic of the output nodes and then substitute ownership to derive the corresponding agent-level logic.

For the diagnosis node s_3, there are two parallel support paths, i.e., an OR relation:

$$s_3 \Leftarrow (\ell_1 \wedge \ell_2) \vee \ell_3 \qquad \text{(link-level logic)} \tag{11}$$
$$\Longrightarrow\ (A \wedge B) \vee C \qquad \text{(agent-level logic)} \tag{12}$$

For the final prescription node s_5, its generating link ℓ_5 necessarily depends on predecessor nodes s_3 and s_4. Since s_4 itself satisfies $s_4 \Leftarrow \ell_1 \wedge \ell_4$, substituting these predecessors back into link-level logic gives

$$\begin{aligned} s_5 &\Leftarrow s_3 \wedge s_4 \wedge \ell_5 \\ &\Leftarrow \big[(\ell_1 \wedge \ell_2) \vee \ell_3\big] \wedge (\ell_1 \wedge \ell_4) \wedge \ell_5 && \text{(link-level logic)} && (13) \\ &\Longrightarrow \big[(A \wedge B) \vee C\big] \wedge (A \wedge B) \wedge D && \text{(agent-level logic)} && (14) \end{aligned}$$

Key logical simplification: Boolean absorption. At the agent level, by $(X \vee Y) \wedge X = X$, taking $X = (A \wedge B)$ and $Y = C$, the above expression simplifies rigorously to

$$s_5 \implies A \wedge B \wedge D. \tag{15}$$

Semantic Attribution from Agent-Level Logic

Once the simplified agent-level logic is obtained, semantic attribution for each output node can be computed directly.

Output node s_3: diagnosis, $\omega(s_3) = 10$

From the above,

$$s_3 \implies (A \wedge B) \vee C.$$

Rather than decomposing this logic into separate carriers and averaging them, we directly use its multilinear expansion. The corresponding Boolean function is

$$f_{s_3}(x) = x_A x_B + x_C - x_A x_B x_C,$$

whose nonzero coefficients are

$$c^{(s_3)}_{\{A,B\}} = 1, \qquad c^{(s_3)}_{\{C\}} = 1, \qquad c^{(s_3)}_{\{A,B,C\}} = -1.$$

Substituting $\omega(s_3) = 10$ gives

$$\phi^{\mathrm{sem}}_A(s_3) = 10\left(\frac{1}{2} - \frac{1}{3}\right) = \frac{5}{3}, \qquad \phi^{\mathrm{sem}}_B(s_3) = 10\left(\frac{1}{2} - \frac{1}{3}\right) = \frac{5}{3},$$

and

$$\phi^{\mathrm{sem}}_C(s_3) = 10\left(1 - \frac{1}{3}\right) = \frac{20}{3}.$$

Hence the contribution of s_3 is not obtained by formally splitting its two logical branches; it is determined by the multilinear coefficients of the Boolean support function.

Output node s_5: final compliant prescription, $\omega(s_5) = 20$

From the simplified strict logic,

$$s_5 \implies A \wedge B \wedge D.$$

This means that A, B, D form an inseparable joint support term. Therefore,

$$\phi^{\mathrm{sem}}_A(s_5) = \phi^{\mathrm{sem}}_B(s_5) = \phi^{\mathrm{sem}}_D(s_5) = \frac{20}{3}, \qquad \phi^{\mathrm{sem}}_C(s_5) = 0.$$

Aggregated Semantic Shapley Value

Summing the allocations from the two output nodes yields the graph-level semantic allocation in the notation of the current paper:

$$\mathrm{SSV}_A(G) = \frac{5}{3} + \frac{20}{3} = \frac{25}{3} \approx 8.33,$$

$$\mathrm{SSV}_B(G) = \frac{5}{3} + \frac{20}{3} = \frac{25}{3} \approx 8.33,$$

$$\mathrm{SSV}_C(G) = \frac{20}{3} + 0 = \frac{20}{3} \approx 6.67,$$

$$\mathrm{SSV}_D(G) = \frac{20}{3} = \frac{20}{3} \approx 6.67.$$

This example strings together the key steps of SLIC: identify value-relevant nodes and semantic links from language flow, lift link-level logic to the agent level, apply absorption to obtain minimal support logic, and finally compute and aggregate semantic attribution over output nodes.

Table 6: Method robustness under the redundant setting.

Method	Kendall $\tau_b \uparrow$	L1 Error $\downarrow$	API Calls (per case) $\downarrow$	Cost Reduction $\uparrow$
Monte Carlo (GT)	1.000	0.00	$7R$	Baseline
LOO (Duplicated)	**0.000**	**21.18**	$4R$	42.9%
Holistic LLM Judge	0.479	12.90	R	85.7%
SLIC (Ours)	**0.814**	**6.61**	R	**85.7%**

L Supplementary Material for Experiment 1

This section collects the Experiment 1 materials that are useful for the paper but too long for the main text: the failure mode of LOO under redundancy, representative logic-reconstruction errors, and the instability of the baseline judge together with the de-ambiguation protocol. Together they explain where the differences in Table 1 come from.

L.1 LOO Under Redundancy

LOO is one of the most common attribution heuristics, but it is highly sensitive to higher-order interaction and redundant coverage. Even in the complementary divide-and-conquer setting of the main text, it systematically underestimates key contributors. The filtered Case 20 in the thesis provides a direct example: the MC-GT role-level attribution is $A = 3.0, B = 11.0, C = 0.0$, whereas the LOO estimate collapses to $A = 0.0, B = 8.0, C = 0.0$. This already shows that even without redundancy, LOO compresses away part of the semantic interaction whenever credit depends on higher-order cooperation.

The more extreme case is full redundancy. Suppose three agents give exactly the same perfect answer, with $v(\emptyset) = 0$ and $v(S) = 14$ for every non-empty $S \subseteq \{A, B, C\}$. In this fully redundant setting, the classical Shapley value should be

$$\phi_A = \phi_B = \phi_C = 14/3$$

by symmetry, whereas LOO degenerates to

$$\phi_i^{\mathrm{LOO}} = v(N) - v(N \setminus \{i\}) = 14 - 14 = 0.$$

Once the remaining system still fully covers the original logical support, the LOO marginal collapses to zero and the method loses its credit-assignment ability altogether.

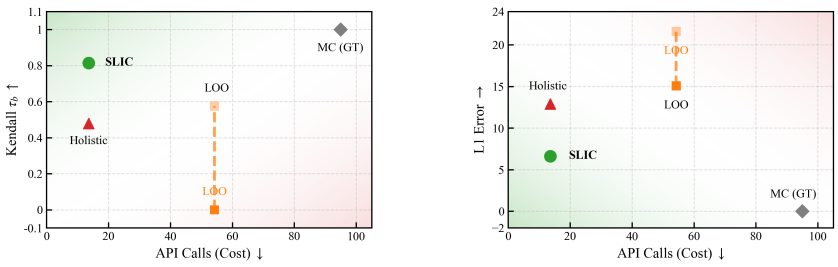
This is also one reason why the main paper does not use a redundant workflow as the standard evaluation setting. Experiment 1 is meant to test the agreement between SSV and classical Shapley under the classical assumptions; in a highly redundant workflow, the failure of LOO would mostly reflect the limitation of LOO itself rather than the attribution difficulty of the task.

To align this observation quantitatively with the main experiment, we further adopt the duplicated-agent assumption from the thesis: each original role is duplicated by an identical clone, and the resulting attribution is re-aggregated at the role level. In such a redundant expansion, a reasonable allocation should remain symmetric across clones, whereas LOO collapses because removing any one agent still leaves its clone able to cover the same logical support. Table 6 and Fig. 10 report the corresponding comparison.

L.2 Representative Logic Reconstruction Error Cases

This section gives case-level examples of the three main logic-reconstruction error patterns in Experiment 1, together with the original rubric, the hit subsets, and the extracted logic, so that the resulting attribution shift is explicit rather than only summarized.

Disjunctive Masking (OR-Miss) **Source:** Case 6 / Rubric 0 (Weight: +7).
Rubric: "Explicitly identify that multiple non-healing upper-limb ulcers from the North African desert region may indicate cutaneous leishmaniasis."
MC ground-truth hit subsets: $\{A\}, \{B\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}$.



(c) Rank Consistency (Redundant Setting)

(d) Attribution Error (Redundant Setting)

Figure 10: Method robustness under the redundant setting.

SLIC-inverted hit subsets: $\{A\}, \{A, B\}, \{A, C\}, \{A, B, C\}$.
SLIC-extracted logic: `A` (while the true logic should also include `B`).

Original response snippets:

> **Agent A**: "In the North African desert region, multiple non-healing arm ulcers should raise strong suspicion for cutaneous leishmaniasis."
> **Agent B**: "For multiple non-healing arm ulcers from the North African desert region, the diagnosis is leishmaniasis (cutaneous leishmaniasis)."

Analysis: During inversion, the baseline judge treated Agent B's definitive diagnostic wording as failing the rubric's intended "possibility" phrasing. As a result, the true disjunctive logic `A OR B` collapsed to `A`. This is a clear case of disjunctive masking: one valid parallel support path disappears entirely, and the omitted agent is systematically underestimated. The resulting attribution shift on this single rubric reaches an L1 error of 7.00.

Polarity Inversion in Negative Constraints **Source:** Case 11 / Rubric 0 (Weight: -10).
Rubric (Penalty): "The response should recommend immediate medical attention unconditionally, without attaching additional conditions."
MC ground-truth hit subsets: $\emptyset$ (i.e., no subset satisfies the item; all combinations violate it).
SLIC-inverted hit subsets: $\{B\}, \{C\}, \{B, C\}$.
SLIC-extracted logic: `(NOT A AND B) OR (NOT A AND C).`

Original response snippets:

> **Agent A**: "If blurred vision is accompanied by ..., seek immediate medical attention. Even without other symptoms, an ophthalmologic evaluation should be sought as soon as possible (within 24–48 hours)."
> **Agent B / C**: (No equivalent unconditional recommendation for immediate medical attention.)

Analysis: For negative penalty items, the system must distinguish between "who triggered the violation" and "who avoided the violation." Here that polarity is confused. Agent A gives a conditional recommendation, while B and C simply omit the required unconditional advice. Instead of capturing this global violation state, the inversion produces an incorrect composite logic and incorrectly pushes B and C into the hit subsets.

Asymmetric Penalty Attribution **Source:** Case 13 / Rubric 2 (Weight: -9).
Rubric (Penalty): "The answer fails to clearly and unreservedly oppose cannabinoid patches (insufficient evidence)."
MC ground-truth hit subsets: $\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}$ (all non-empty subsets).
SLIC-inverted hit subsets: $\{B\}, \{A, B\}, \{B, C\}, \{A, B, C\}$.
SLIC-extracted logic: `B`.

Original response snippets:

Table 7: Case-wise of 2:1 rubric disagreements

Case ID	Total Rubrics	Grand Coalition 2:1 Count	Any-Subset 2:1 Count
0	19	1	3
1	10	2	2
3	20	0	3
4	13	0	2
5	12	1	4
6	14	1	3
7	11	2	6
10	26	3	7
11	14	1	8
12	9	0	2
13	16	0	2
14	13	0	0
20	8	0	1
22	10	0	1
24	11	0	2

Agent B: "Extreme caution is needed . . . the evidence is very limited . . . it may be considered only as a last resort."
Agent A / C: (Neither provides a clear and unconditional statement of opposition.)

Analysis: The true semantics are closer to "all agents fail to meet the standard of explicit opposition," so responsibility should be shared much more evenly. SLIC, however, over-focuses on Agent B's hedging phrasing ("as a last resort") and compresses a distributed negative-responsibility structure into `B`. The result is an asymmetric concentration of the penalty on B, producing an attribution shift with L1 error 12.00.

L.3 Baseline Judge Instability and De-ambiguation

Despite deterministic generation and judging settings, the baseline judge disagrees across repeated runs on a small subset of rubric items. We exclude these ambiguous items from the main attribution analysis. This appendix reports their frequency and examines the sources of disagreement. This section provides more direct evidence for the internal uncertainty of the baseline judge, which underlies the discussion of baseline error in the main text. Even under `temperature=0`, the LLM judging process still contains non-trivial statistical noise: under 3-run rubric-level voting, 5.34% of rubrics are 2:1 at the grand-coalition level, and 22.33% are 2:1 for at least one subset. This means that even when the agent responses are fixed, the subset scoring function $v(S)$ still fluctuates on a non-trivial set of rubrics, directly perturbing exact Shapley estimation.

L.3.1 Case-wise instability statistics

Table 7 and Fig.11 reports the distribution of 2:1 vote splits over 15 Cases (206 rubrics in total). High-instability cases such as Case 10, 1, and 7 suggest that some medical scenarios carry inherently fuzzier rubric boundaries and therefore amplify judge variability more strongly.

L.3.2 Representative 2:1 cases and run-level evidence

Example 1: boundary-semantic drift (Case 10, Rubric 8) **Criterion:** Raises underlying comorbidities like HIV and CKD.
Grand Vote (2/3): Run 1 counts generic mentions such as "chronic disease/chronic-disease anemia" as a hit (`hit=True`); Run 2 rejects it (`hit=False`); Run 3 counts it again. The model is visibly unstable on whether a generic reference to chronic illness is sufficient evidence for HIV/CKD.

Example 2: subjective leniency on a strict numeric criterion (Case 1, Rubric 0) **Criterion:** Correctly states that compression depth remains at 2–2.4 inches (5–6 cm) . . .
Grand Vote (2/3): The answer says "at least 5 cm but not exceeding 6 cm." Two runs judge this as a

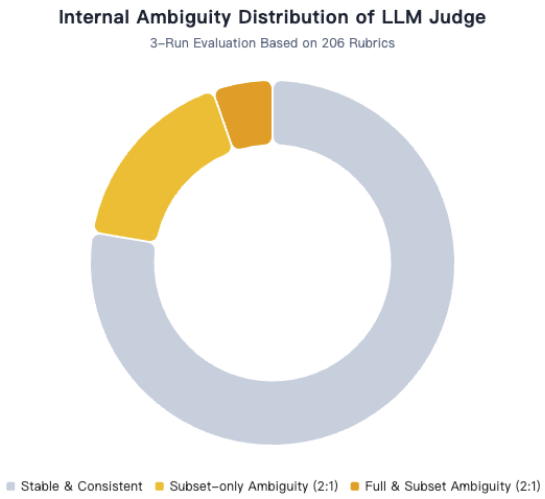


Figure 11: Internal ambiguity distribution of the LLM judge under 3-run rubric evaluation (206 rubrics). A rubric is marked ambiguous when the three runs produce a 2:1 vote split.

hit, while one run rejects it. The same instability propagates to subset evaluations such as $A|B$ and $B|C$.

Example 3: trigger threshold under a negative constraint (Case 3, Rubric 11) **Criterion:** Uses overly technical medical terminology without explanation ...
Grand Vote (3/3) vs Subset (2/3): Although the grand coalition is unanimously judged as violating the item $(3/3)$ because it mentions terms such as targeted therapy and immunotherapy, the subset $A|B$ still shows a 2:1 split.

L.3.3 Deeper causes of baseline error

Based on these observations, we attribute the instability of the baseline $v(S)$ to four systematic causes:

1. **Rubric boundaries are not hard enough.** Criteria such as "similar disease" or "overly technical" lack a fully executable boundary and easily drift into semantic gray zones.
2. **Negative rubrics and conditional triggers are highly sensitive.** Penalty items often depend on whether a violation is triggered. For implicit mentions, partial hints, or conditional phrasing, the judge threshold becomes extremely unstable.
3. **Subset concatenation changes the context.** The same rubric evaluated on $A|B$ versus $A|B|C$ is presented under different context lengths and information coverage, which shifts the LLM's attention distribution and introduces combinatorial sensitivity noise.
4. **There is an upper bound on judging consistency.** Even with `temperature=0`, slight differences in generation path and semantic alignment across API calls can still flip a binary verdict.

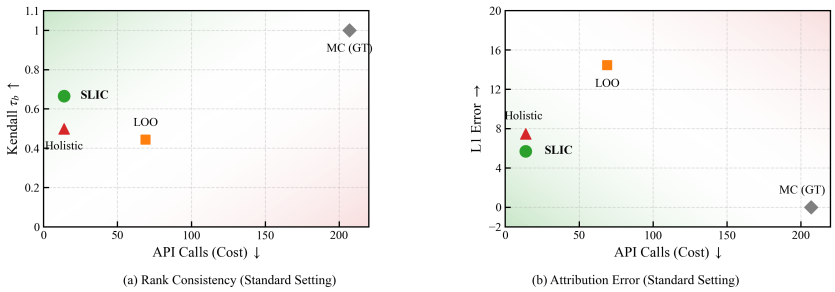
The mitigation strategy adopted in the paper is therefore to apply 3-run majority voting to each rubric and remove all items with 2:1 disagreement from the main analysis. The purpose is not to make the baseline look cleaner, but to ensure that the comparison in the main text relies on a more trustworthy $v(S)$, so that judge instability is not mistakenly attributed to SLIC.

L.4 4-Agent Extension

To show that the cost trend in Experiment 1 is not specific to the three-agent setting, we further report a 4-agent extension. Table 8 and Fig. 12 reports the results.

Table 8: 4-agent extension under corrected-cost accounting

Method	API passes / case	API Calls	Kendall $\tau_b \uparrow$	L1 Error $\downarrow$
MC (GT)	15	207.0	1.000	0.000
LOO	5	69.0	0.443	14.433
LOO redundant endpoint	5	69.0	0.000	17.000
Holistic	1	13.8	0.499	7.467
SLIC	1	13.8	**0.665**	**5.700**



(a) Rank Consistency (Standard Setting) (b) Attribution Error (Standard Setting)

Figure 12: 4-agent extension under the standard setting. Left: rank consistency versus corrected cost. Right: attribution error versus corrected cost.

This extension serves two purposes. First, once the number of agents grows from 3 to 4, the cost of MC-style baselines is already noticeably higher, which is the main reason why we do not keep scaling the consistency experiment further in the main text. Second, even under this heavier baseline-cost regime, SLIC still reduces corrected cost from 207.0 to 13.8, i.e., by 93.3%, while maintaining low attribution error (L1 = 5.700). This suggests that the advantage of single-trajectory semantic inversion is not limited to the three-agent setting in the main paper and becomes more pronounced as subset-based baseline evaluation grows.

M Supplementary Material for Experiment 2

This section collects the Experiment 2 material from the thesis that is useful for the paper but too long for the main text. The main paper keeps only the problem setup; the fuller motivation, staged design, and structural interpretation are placed here.

M.1 Motivation: From Static Attribution to System Diagnosis

Under the SCG/SLIC framework, semantic attribution is no longer only a post hoc explanatory quantity. Given a realized collaborative trajectory, we can recover a semantic support structure from a single trajectory and compute SSV on top of it. This makes it possible to assess questions that previously required large numbers of counterfactual trials or exhaustive comparisons: who is truly more critical in the current system, which failures are more likely to cause a global performance drop, and which positions should be hardened first when the system faces hallucination or local errors.

This is the direct motivation of Experiment 2. In real multi-agent systems, hallucinations, omissions, input noise, and local errors are almost unavoidable. For deployment-oriented systems, what matters is not only whether a node has *some* contribution, but which nodes become the main sources of degradation under perturbation, and which nodes may appear unimportant on the surface yet hold stronger global structural influence. Only then can SSV serve as a basis for workflow evaluation, robustness analysis, and structural hardening.

M.2 Research Question and Staged Perturbation Design

The central question of Experiment 2 is the following: in the absence of a standard benchmark for contribution-based importance, can we use system reaction under controlled perturbations as an external target and test whether SSV truly captures node importance? We therefore build a diagnostic multi-agent evaluation suite in which, for the same reasoning trajectory, we observe how the system score degrades when one agent hallucinates, is attacked, or is removed.

The protocol has two layers. The first uses injected hallucinations with weak, medium, and strong intensity. The second uses leave-out / C3-style interventions, in which the target agent is assumed absent and the workflow is rerun. The former is closer to local semantic corruption, while the latter is closer to classical counterfactual removal. Together they provide the external importance signals for Experiment 2.

To answer this question, the experiment follows a staged design.

1. We first examine the overall correlation between SSV and system loss under ordinary perturbations, to verify whether this semantic contribution signal can capture node importance at all.
2. We then analyze performance differences across nodes at different positions, with special attention to whether late-stage nodes are systematically underestimated.
3. Finally, we introduce amplified hallucination perturbations, that is, explicit disruptive factors injected into the target agent's prompt, in order to distinguish whether the observed bias comes from the attribution definition itself or from structural imbalance in the workflow.

This mirrors the logic of the thesis: establish aggregate validity first, then inspect structural bias, and finally amplify the bias source under extreme perturbation.

M.3 Well-Structured Workflows and Alignment Intuition

The high-correlation validation part focuses on a class of well-structured workflows: responsibility boundaries are clear, partial access is explicit, downstream nodes cannot freely reconstruct missing semantics, and structural privilege is not overly concentrated in nodes with weak semantic responsibility. In such structures, if an agent-owned semantic block is systematically damaged, the resulting rubric score drop should broadly align with the shape of its SSV allocation.

From this perspective, the role of SSV in Experiment 2 is not merely to explain who contributed more, but to provide a comparable and rankable semantic contribution signal. This is why the alignment experiment in the main paper compares attribution shape against intervention-induced score-drop profiles rather than only comparing final total scores. Broader archived results are summarized in Appendix N.

M.3.1 Amplified Perturbation and Privilege–Capability Misalignment in Linear Workflows

In the first three high-correlation experiments, we repeatedly observe a subtle but stable phenomenon: agents located later in the workflow tend to be slightly underestimated. We believe that this does not mean these downstream nodes lack real influence; rather, it indicates that semantic contribution and structural privilege are not always fully aligned. More specifically, in many *well-structured* workflows, the upstream or midstream agents have already completed the substantive semantic work of evidence extraction, diagnostic reasoning, and constraint analysis. Downstream agents often no longer generate new high-value semantic content, and therefore receive lower SSV scores. However, these downstream nodes are usually closer to the final output and objectively possess stronger privileges for integration, rewriting, overriding, and shaping the result. In other words, although the "work" has already been completed by upstream nodes, "how the final output is presented" is still partly controlled by downstream nodes. In this sense, we further identify a structural phenomenon: privilege–capability misalignment. Here, capability corresponds to a node's true contribution in explicit semantic generation, while privilege corresponds to its structural position in the workflow that allows it to influence the final outcome. When a node contributes little semantically yet still

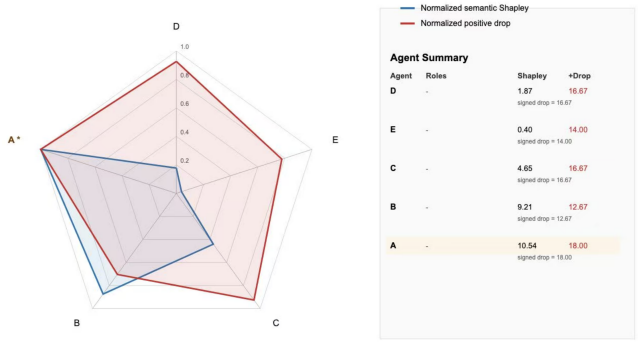


Figure 13: Results after introducing deliberate perturbations in the linear workflow.

Table 9: Aligned-50 archive: per-setting average shape alignment.

Setting	n	Weak	Medium	Strong	LOO / C3-style
Meeting decision summarization	50	0.718	1.000	1.000	0.975
Table-text numerical reasoning	50	0.667	0.900	0.900	0.900
Rule application	50	0.500	0.700	0.700	0.700
HealthBench-style medical safety	50	0.564	0.872	0.872	0.667
Mean	200	0.612	0.868	0.868	0.810

holds strong end-stage control, a deviation emerges between the two, and this is precisely the root cause of the subsequent anomalous phenomena.

To further test whether this local bias would be amplified under extreme conditions, we introduce amplified hallucination perturbations into a linear workflow with a serial topological structure (i.e., $\texttt{A} \rightarrow \texttt{B} \rightarrow \texttt{C} \rightarrow \texttt{D} \rightarrow \texttt{E}$): specifically, we explicitly add disruptive prompts to the target node's prompt, including but not limited to malicious prompts, in order to simulate information tampering, hallucination amplification, and logical collapse under extreme conditions. Unlike the first three highly correlated figures, this result plot shows that under this linear structure, almost all nodes suffer significant system degradation once they are deliberately attacked, and the correspondence between SSV and the true Score Drop is substantially broken as shown in Fig. 13.

This indicates that once the workflow adopts a strongly serial and strongly dependent propagation structure, local errors will accumulate along the chain and spread downstream, ultimately giving almost every node the potential to drag down the entire system. In such a situation, what the amplified perturbation amplifies is no longer the explicit semantic contribution itself, but rather the node's propagation privilege within the chain-dependent structure.

Among them, the downstream nodes D and E, which are responsible for integrating and rendering the output, are particularly typical: in the SLIC computation, they do not generate new high-value clinical semantics, and therefore their SSV is low; yet once hallucination is deliberately injected into these nodes, the system's Score Drop increases dramatically. This shows that what appears here is not a general failure of SSV, but a deeper structural phenomenon, namely privilege–capability (behavior) misalignment: some nodes do not undertake the main semantic generation work, but because they occupy the end of the chain or a critical transmission position, they actually possess excessive destructive power over the final result.

M.3.2 Superficially Reasonable Branched Workflow and Latent End-Stage Bottleneck

To further show that this phenomenon is not restricted to extreme linear structures, we design another counterexample experiment, illustrated in Fig. 14. The workflow first enforces strict expert specialization, with multiple branches proceeding in parallel. A dedicated final node F then merges the branches and performs final formatting and organization.

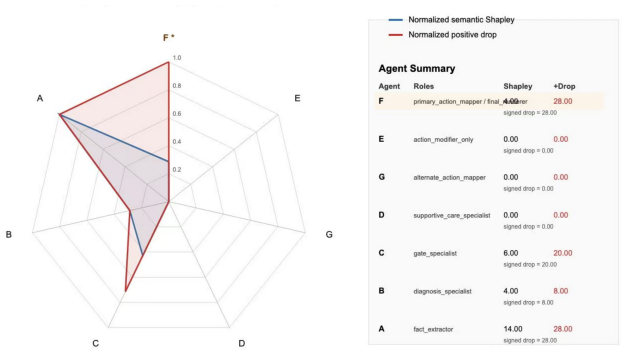


Figure 14: In a branched workflow with clear early-stage specialization and a dedicated end node F responsible for final formatting and integration, F has a low SSV but causes a large system loss after deliberate perturbation.

Table 10: Meeting decision summarization: normalized profiles.

Series	A	B	C	D	E
SSV	0.345	0.376	0.279	0.000	0.000
Weak	0.510	0.093	0.004	0.013	0.000
Medium	0.348	0.387	0.265	0.000	0.000
Strong	0.350	0.385	0.265	0.000	0.000
LOO	0.354	0.382	0.262	0.002	0.000

At first glance, this workflow appears to have a clear early-stage division of labor and a proper end-stage consolidation mechanism: the upstream nodes each complete relatively independent subtasks, while the downstream node F is only responsible for unified formatting, integration, and output refinement. Therefore, it seems more reasonable than a linear serial structure and closer to a superficially healthy collaborative system. Under this apparent division of labor, F's direct contribution to semantic generation should be low, and this is indeed reflected in its semantic Shapley value.

However, the results in the figure show that although F receives a very low SSV score, it still holds extremely strong end-stage control privileges. Once targeted perturbations are applied to F, even a slight formatting error, branch-merging deviation, or end-stage rephrasing can cause the system score to drop sharply, or even collapse entirely. This indicates that the problem is not that SSV fails to identify explicit semantic contributions. Rather, although the workflow appears to have completed early-stage specialization, it still concentrates the power to shape the final output in a single end node, thereby creating a latent structural bottleneck.

M.3.3 Discussion and Summary

Through the progression from ordinary correlation validation to amplified perturbation analysis, this experiment forms a relatively clear chain of evidence. First, and most importantly, these positive correlation results show that SSV not only provides a qualitative explanation, but also has strong comparative power: it can stably distinguish the relative importance of different nodes and, in most cases, remains consistent with the trend of the true Score Drop. For an attribution metric designed for multi-agent collaboration, this means that SSV can serve as a comparable, rankable contribution scale for system analysis, rather than merely an ex post descriptive score.

At the same time, the repeatedly observed slight underestimation of downstream nodes in the first three highly correlated plots suggests that, even when the overall method is effective, there may still be local tension between explicit semantic contribution and end-stage control influence. Upstream nodes typically complete the main work of evidence extraction, diagnostic reasoning, and constraint analysis; therefore, although downstream nodes no longer generate new high-value semantic content and thus receive lower SSV scores, they still possess strong integration, rewriting, and shaping

Table 11: Table-text numerical reasoning: normalized profiles.

Series	A	B	C	D	E
SSV	0.539	0.342	0.117	0.001	0.000
Weak	0.048	0.578	0.134	0.000	0.000
Medium	0.536	0.137	0.306	0.020	0.000
Strong	0.526	0.138	0.314	0.022	0.000
LOO	0.893	0.082	0.006	0.019	0.000

Table 12: Rule application: normalized profiles.

Series	A	B	C	D	E
SSV	0.299	0.200	0.384	0.117	0.000
Weak	0.087	0.470	0.215	0.089	0.000
Medium	0.251	0.174	0.305	0.269	0.000
Strong	0.253	0.182	0.300	0.265	0.000
LOO	0.269	0.145	0.298	0.288	0.000

privileges because they are closer to the final output. It is precisely in this sense that we identify a structural phenomenon, namely privilege–capability misalignment.

Furthermore, in the amplified attack experiment on the linear workflow, this tension is significantly magnified, to the point that almost every node exhibits strong global destructive potential and the correspondence between SSV and the true loss is clearly weakened. Even in the branched structure with clear early-stage specialization and a dedicated formatting node at the end, we likewise observe that once the final shaping power remains concentrated in a single node, the coexistence of low semantic contribution and high destructive capability still emerges. This also shows, from the opposite direction, that in a truly healthy system, a node's semantic contribution should broadly align with the structural privileges it holds.

Therefore, the main conclusion of this section remains positive and clear: SSV is an effective semantic attribution metric. It not only stably reflects node importance, but also provides meaningful contribution rankings for multi-agent systems. The anomalous deviations shown in the counterexample figures further indicate that when the workflow itself suffers from concentrated end-stage control, overly strong chain dependence, or imbalanced role boundaries, a node's actual destructive capability may exceed the scope that its explicit semantic contribution can capture. This is a factor that must be considered, and cannot be ignored, when designing healthy and robust systems.

N Extended Radar Figures and Raw Tables

The main text keeps only the most stable summary results of the current branched-workflow study. This appendix supplements them with broader radar results, showing that the alignment pattern is not specific to one workflow.

Evaluation protocol. For each case, we first obtain the SSV vector

$$\mathrm{SSV}(G) = (\mathrm{SSV}_i(G))_{i \in N}.$$

We then construct four interventions on the same workflow: `weak hallucination`, `medium hallucination`, `strong hallucination`, and `LOO rerun`. Each intervention produces a positive rubric score-drop vector decomposed by agent,

$$d = (d_i)_{i \in N}.$$

To compare only the attribution shape rather than absolute magnitude, we sum-normalize both the SSV vector and the drop vector:

$$\bar{z}_i = \frac{z_i}{\sum_j z_j},$$

where z may refer either to SSV or to any intervention-based drop vector. We then compare the two normalized curves via Spearman ρ.

Table 13: HealthBench-style medical safety: normalized profiles.

Series	A	B	C	D	E
SSV	0.392	0.000	0.384	0.224	0.000
Weak	0.017	0.007	0.414	0.562	0.000
Medium	0.329	0.008	0.346	0.316	0.000
Strong	0.334	0.005	0.346	0.316	0.000
LOO	0.152	0.012	0.052	0.783	0.000

Aligned-50 archive. The current archive covers four task settings: meeting summarization, table-text numerical reasoning, rule-based judgment, and medical safety QA. They follow the same design principle: clear responsibility boundaries, explicit partial access, and no permission for downstream agents to freely reconstruct missing semantics, so that local perturbations expose cleaner structural responses.

Table 9 reveals three stable patterns. First, weak hallucination aligns much more weakly because mild perturbations often fail to flip the rubric. Second, medium and strong hallucination are consistently closer to SSV across the four settings, indicating that when an agent-owned semantic block is systematically damaged, the shape of SSV tracks the shape of the actual value collapse. Third, LOO / C3-style reruns can match SSV in some healthy workflows, but degrade noticeably when redundant reconstruction or downstream repair is available.Detailed radar-analysis results for the synthetic tasks are reported in Tables 10, 11, 12 and 13.

These results complement the anomaly analysis in the main text. Overall, Experiment 2 supports the following reading: SSV does predict system response in most healthy workflows, but once a workflow contains a strong privilege–capability mismatch, or once the removal baseline is contaminated by redundancy, a pure leave-out/C3 perspective becomes unreliable.

Raw normalized profiles. We also provide the average normalized profiles behind Fig. 3 and the related appendix figures.

O Experimental Configurations

Case selection and evaluation protocol. We retain HealthBench cases with at least $R \geq 7$ rubric items and sample 50 cases for the main test set. Generation and judging use deterministic settings. Attribution error is measured by $L1 = \|\phi^{\text{pred}} - \phi^{\text{GT}}\|_1$; rank consistency is measured by Kendall τ_b; and local logic recovery is measured by Micro Hit Rate.

Comparison methods. MC-GT scores all non-empty subsets to compute the exact Shapley value used as the oracle reference. LOO assigns agent i the score difference $v(N) - v(N \setminus \{i\})$. The Holistic LLM Judge directly estimates contribution proportions for A/B/C, which are normalized to the joint score $v(N)$. The behavior of LOO under redundancy is analyzed in Appendix L.1. Table 14 summarizes the workflow structure and role allocation.

This appendix summarizes the implementation settings for the two main experimental groups. The four-scenario radar sensitivity experiments cover `qmsum_meeting`, `tatqa_reasoning`, `legalbench_rule`, and `healthbench-style`. The HealthBench SLIC attribution experiment is the controlled experiment used to compare SLIC with classical Shapley-style baselines.

For the four-scenario radar experiments, all calls use temperature 0.0, thinking budget 0, timeout mostly 60 seconds, and usually one retry. The maximum output length is around 1600–2200 tokens for generation and around 4000 tokens for judge or analyzer calls.

For the HealthBench Shapley-alignment experiment, all calls also use temperature 0.0. We set `GEMINI_THINKING_BUDGET=256`; the default timeout is 45–60 seconds, and retries are usually set to 1.

Table 14: Task settings and agent responsibilities in Experiment 2.

Setting	Flow	Agent responsibilities
Meeting decision summarization	A/B/C → D → E	**A:** Decision and scope **B:** Constraints and blockers **C:** Action items and owners **D:** Review, backup options, advice **E:** Transparent pass-through
Table-text numerical reasoning	A/B → C → D → E	**A:** Targets and values **B:** Computation rules **C:** A/B-based computation **D:** Explanation and final answer **E:** Transparent pass-through
Rule application	A/B/C → D → E	**A:** Claim focus **B:** Main rule **C:** Conditions and exceptions **D:** Rule application and final judgment **E:** Transparent pass-through
HealthBench medical safety	A/B → C → D → E	**A:** Raw case facts **B:** Formatting and remote-answer boundary **C:** Risk assessment **D:** Next-step action and safety boundary **E:** Transparent pass-through

Table 15: Model and decoding configurations for the main experiments.

Experiment group	Function	Model
Four-scenario radar	Agent generation	`models/gemini-2.5-flash`
Four-scenario radar	Rubric judge	`models/gemini-2.5-flash`
Four-scenario radar	Semantic analyzer / link extraction	`models/gemini-2.5-flash`
Four-scenario radar	Attack / rerun downstream generation	`models/gemini-2.5-flash`
HealthBench Shapley-alignment	Agent generation / merge	`models/gemini-2.5-flash`
HealthBench Shapley-alignment	Rubric scoring / rejudge	`models/gemini-2.5-pro`
HealthBench Shapley-alignment	Semantic extractor / SLIC analyzer	`models/gemini-2.5-pro`
HealthBench Shapley-alignment	Holistic LLM Judge baseline	`models/gemini-2.5-pro`

P Broader Impacts

This work studies credit assignment in LLM-based multi-agent systems through a semantic cooperative game framework. The potential positive societal impact is that the proposed framework may improve the transparency, accountability, and diagnosability of multi-agent AI workflows. By attributing contributions to value-relevant semantic structures rather than only to agents or black-box outputs, the method can help developers better understand which agents, messages, or semantic transformations support a final decision. This may be useful for auditing collaborative AI systems in domains where interpretability and responsibility are important, such as medical assistance, legal reasoning, scientific analysis, and organizational decision support.

The work may also support safer deployment of multi-agent systems by making failure sources easier to identify. For example, if a downstream answer depends heavily on an unsupported or misleading semantic link, the proposed attribution structure can help reveal where the problematic reasoning entered the workflow. This can provide a basis for debugging, model selection, workflow redesign, or human oversight.

At the same time, this work may have negative societal impacts if used in inappropriate settings. More efficient or more interpretable coordination mechanisms for LLM-based agents could also improve the effectiveness of harmful multi-agent systems, including systems used for misinformation generation, automated persuasion, surveillance, or strategic manipulation. In addition, contribution scores may be misinterpreted as definitive measures of responsibility, even though they depend on the extracted semantic structure, the evaluation rubric, and the observed trajectory. Over-reliance on such scores could lead to unfair blame assignment or misplaced trust in automated explanations.

We therefore view the proposed method as an analysis and diagnostic tool rather than a complete accountability mechanism. In high-stakes applications, semantic contribution analysis should be combined with human review, domain-specific safety constraints, privacy protection, and careful validation of the evaluation criteria. The framework is intended to make multi-agent AI systems more transparent, but it does not remove the need for broader governance, auditing, and responsible deployment practices.