

Supplementary Material for

Epistemic Topology for AI Agent Systems

Bogdan Banu
bogdan@banu.be

April 22, 2026

A Detailed Proof Sketches

This appendix collects the longer derivation steps omitted from the main text. The goal is not to expand the model beyond the submission claim, but to show how the topology-level formulas in Section 5 follow from the visibility assumptions.

A.1 Reviewer Gates and Correlated Failure

Let $X = \mathbf{1}_{E_{\text{gen}}}$ and $Y = \mathbf{1}_{M_{\text{ver}}}$. In the direct topology, failure is exactly E_{gen} , so

$$P(\text{Fail}_{\text{direct}}) = P(E_{\text{gen}}) = p.$$

In the reviewer-gated topology, failure is the conjunction $E_{\text{gen}} \wedge M_{\text{ver}}$. Under independence this gives

$$P(\text{Fail}_{\text{gate}}) = pq.$$

For the correlated case,

$$P(E_{\text{gen}} \wedge M_{\text{ver}}) = \mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \text{Cov}(X, Y) = pq + \rho\sqrt{p(1-p)q(1-q)}.$$

The topology contributes the conjunction; implementation choices determine the covariance term.

A.2 Error Amplification

Let E_j denote the event that worker j emits an erroneous output, with $P(E_j) = p$ independently across workers.

Independent aggregation. The aggregate fails when at least one worker error survives to the combined output. The exact failure probability is

$$P(\text{failure}_{\otimes}) = 1 - (1 - p)^n.$$

Applying the union bound gives

$$P(\text{failure}_{\otimes}) \leq \sum_{j=1}^n P(E_j) = np,$$

so the amplification factor satisfies $A_{\otimes} \leq n$.

Centralized aggregation. Suppose a hub observes all worker outputs before release and detects a worker error with rate $d = P(\text{hub detects error} \mid \text{error occurred})$. A worker error reaches the final output only if it occurs and escapes detection, so the exact centralized failure probability is

$$P(\text{failure}_o) = 1 - (1 - p(1 - d))^n.$$

Again applying the union bound yields

$$P(\text{failure}_o) \leq np(1 - d),$$

hence $A_o \leq n(1 - d)$.

Exact comparison. The ratio

$$\frac{A_\otimes}{A_o} = \frac{1 - (1 - p)^n}{1 - (1 - p(1 - d))^n} \xrightarrow{p \rightarrow 0} \frac{1}{1 - d}.$$

shows that the small- p regime recovers the intuitive $1/(1 - d)$ gain.

A.3 Sequential Handoff Overhead

Let step j be executed by agent a and step $j + 1$ by agent b . The sender’s full task-relevant state is S_a , while the receiver sees only a transmitted summary $\text{obs}_b(S_a)$.

Applying the data processing inequality to the Markov chain $S_a \rightarrow \text{obs}_b(S_a) \rightarrow \hat{r}_j$ gives

$$I(\text{obs}_b(S_a); r_j) \leq I(S_a; r_j).$$

Define the lost task-relevant information as

$$\Delta I_j = I(S_a; r_j) - I(\text{obs}_b(S_a); r_j) \geq 0.$$

Equality holds only if the handoff is lossless for the task-relevant signal. In practical agent systems, finite context windows, summarization, and schema compression make strict equality atypical.

If h handoffs occur in a k -step task, the total multi-agent cost can be written as

$$C_{\text{multi}} = k \cdot c_{\text{step}} + \sum_{j=1}^h (c_{\text{comm}} + c_{\text{recon},j}),$$

where $c_{\text{recon},j}$ is the receiver’s effort to compensate for the missing context. This yields the overhead ratio used in the main text and explains why repeated handoffs on strictly sequential tasks can accumulate superlinearly in practice.

A.4 Parallel Acceleration

Suppose a task decomposes into subtasks $\varphi_1, \dots, \varphi_m$ and that each worker can solve its assigned subtask from local observations alone. Let c_{sub_i} be the wall-clock cost of subtask i .

Parallel cost. When all subtasks run concurrently, total wall-clock time is the slowest subtask plus coordinator assignment and aggregation overhead:

$$C_{\text{parallel}} = \max_i(c_{\text{sub}_i}) + c_{\text{assign}} + c_{\text{agg}}.$$

The single-agent sequential baseline is

$$C_{\text{sequential}} = \sum_{i=1}^m c_{\text{sub}_i},$$

which yields the speedup formula used in the main text.

Coordinator visibility. If the coordinator must receive the tuple of worker outputs (o_1, \dots, o_m) to aggregate them, then for any proposition φ measurable with respect to that tuple, pooled worker-output knowledge implies coordinator knowledge:

$$D_G(\varphi) \Rightarrow K_{\text{hub}}(\varphi).$$

This is why cross-checking comes “for free” in centralized decompositions: the aggregation step already creates pooled visibility.

Epistemic ceiling. If some worker’s optimal action depends on another worker’s hidden local result, then the subtasks are not epistemically independent. Running them in parallel forces a worker to act without a needed fact, which appears in the main text as quality loss proportional to the unresolved mutual information.

A.5 Tool Density

Let T_1, \dots, T_n be an approximately balanced partition of t tools across n agents, so $|T_i| \approx t/n$.

Remote-tool coordination. If agent i needs tool $k \in T_j$ with $j \neq i$, three steps are required:

1. establish that some remote tool can solve the current subproblem;
2. delegate execution to the remote owner;
3. reconstruct and interpret the returned result without the owner’s full local context.

This is the operational content of the coordination term in the main text.

Second-order planning. Planning is no longer just “which tool should I call?” but “which agent knows how to apply which tool to the current subproblem?” In epistemic terms, the planner must reason about propositions of the form $K_i(K_j(\text{tool}_k \text{ can solve } \varphi))$. In the worst case this introduces $O(tn)$ comparisons across tool-agent pairs, versus $O(t)$ for the single-agent baseline.

B Additional Notes on Benchmark Mapping

The benchmark comparison in the main paper is intentionally qualitative. Kim et al. [1] report architecture-level aggregates across heterogeneous tasks, models, and coordination regimes. Those metrics are useful for checking whether the theory predicts the right ordering across topology classes, but they should not be interpreted as direct fitted values of variables such as d , ΔI_j , or S in the simplified formulas of Section 5.

This distinction matters especially for rebuttal. The paper does not claim that the theory numerically reconstructs the benchmark tables. It claims that the benchmark outcomes are directionally consistent with the visibility-based predictions: review bottlenecks reduce error amplification, artificial handoffs hurt strictly sequential tasks, decomposition helps on approximately independent tasks, and large toolsets increase coordination tax in distributed regimes.

References

- [1] Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. Towards a science of scaling agent systems. *arXiv preprint arXiv:2512.08296*, 2025. Google Research, Google DeepMind, and MIT.