WHO ELSE SHOULD SEE IT? GROUP-AWARE EVIDENCE ROUTING FOR MULTI-AGENT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In multi-agent search systems, deciding whether evidence is useful to one agent does not determine how widely it should be shared. We present MERGE (Multi-Agent Evidence Routing with Group Effects), a router that learns how the benefit of sharing evidence with an additional agent depends on how many agents already hold it. Using paired interventions that differ by one additional agent, MERGE combines this learned signal with agent-specific usefulness and communication cost, then selects routing decisions jointly. Across nine QA dataset–backbone settings, MERGE achieves the highest mean F1 and EM scores. On the primary HotpotQA/Qwen2.5-14B setting, it reaches 0.716 F1, improving by 2.4 points over the strongest controlled baseline while using 11.8% fewer tokens. Analysis reveals early diminishing returns from redundant sharing and a late-stage effect tied to the fixed voting rule. Together, these results show that routing quality depends on both how information is distributed across the team and how the team combines its final decisions. Code is available at `https://anonymous.4open.science/r/merge-routing-053E/`.

1 INTRODUCTION

Large Language Model (LLM) agents can search, reason, and exchange information before producing a joint answer (Guo et al., 2024; Du et al., 2023; Liu et al., 2025; Ruan et al., 2026). Communication can improve coordination, but broadcasting every retrieved item to every agent is costly and can fill context windows with redundant evidence. Recent systems therefore prune messages, adapt communication links, or decide which context each agent should receive (Zhang et al., 2025; 2024; Liu et al., 2025; Guan et al., 2026; Hu et al., 2026).

This decision can be viewed as *evidence routing*: deciding which agents should receive each piece of discovered evidence. A natural routing rule is to ask whether the evidence is useful to a particular agent. This local view is important, but it does not determine how widely the same evidence should be shared. Suppose one agent discovers an important fact. Sharing it with another agent may substantially help the team, while giving the same fact to every remaining agent may add little. In another state, however, informing one more agent may change the team's final decision. Thus, the value of sharing evidence with an additional agent can depend on how widely that evidence is already available within the team.

An agent that already has access to a given piece of information is referred to as a *holder*. We refer to the dependence on the number of current holders as a *group effect*. Figure 1 illustrates this distinction. A local-only router evaluates each agent separately, whereas a group-aware router can change its decision as the same evidence reaches more members of the team. Such dependence is relevant in multi-agent reasoning: correlated information can reduce the value of combining multiple views (Pivato, 2017; Bikhchandani et al., 2024), while LLM teams can exhibit reduced diversity, early convergence, and conformity (Chen et al., 2026; Cui et al., 2026; Zhu et al., 2026).

To model this dependence, we introduce MERGE (**M**ulti-Agent **E**vidence **R**outing with **G**roup **E**ffects). The method combines agent-specific usefulness with a learned estimate of how the value of sharing changes as evidence reaches more agents. This estimate is obtained from paired interventions that start from the same intermediate team state. The two runs differ only in whether one additional agent receives the evidence, while following the same downstream communication schedule. Their difference in final team performance provides a controlled signal for the benefit of sharing with that

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.

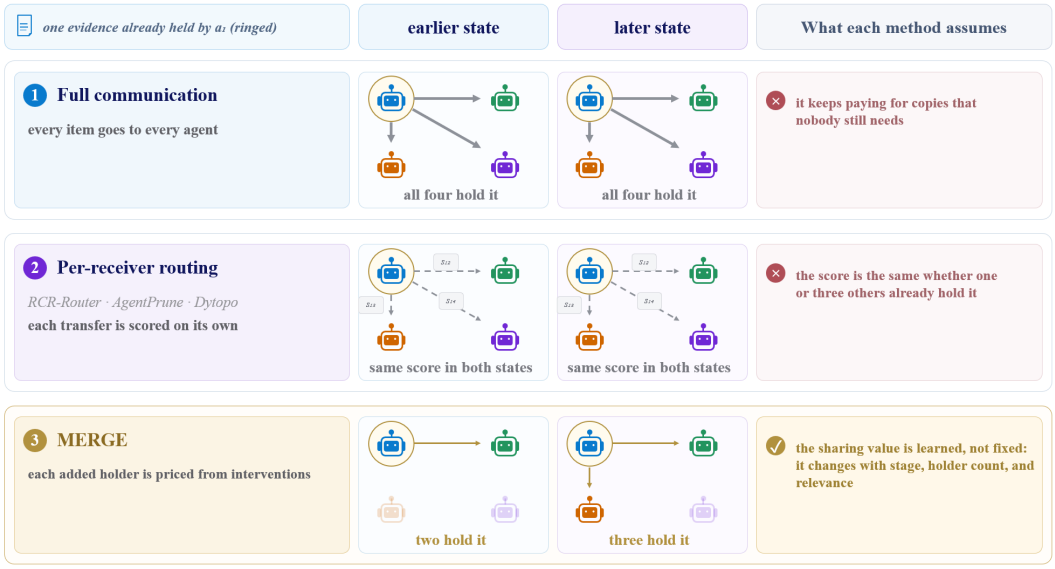


Figure 1: A local-only router evaluates each agent separately. MERGE also models how the value of sharing the same evidence changes as more agents already have access to it.

additional agent. At routing time, MERGE combines this signal with agent-specific usefulness and communication cost to select evidence routing decisions jointly.

Our setting fixes the agent team, retrieval procedure, and final aggregation rule, while allowing the routing policy to determine how discovered evidence is shared across agents.

Experiments cover HotpotQA, MuSiQue, and 2WikiMultiHopQA across multiple LLM backbones, together with transfer to larger teams and HoVer fact verification (Jiang et al., 2020). Across all settings, MERGE achieves the highest mean F1 and EM. On HotpotQA/Qwen2.5-14B, it reaches 0.716 F1, improving by 2.4 points over the strongest controlled baseline while using 11.8% fewer tokens. The analysis further reveals two distinct patterns: broader sharing shows diminishing returns early in search, while a sharp late-stage gain is closely associated with the fixed aggregation rule.

Our main contributions are:

- We formulate evidence routing as a group-aware decision problem: whether evidence is useful to one agent does not determine how widely it should be shared.
- We introduce MERGE, which learns how the benefit of sharing evidence with an additional agent changes with the number of current holders, and uses this signal for joint routing.
- We show consistent gains across datasets, backbones and team sizes, and analyze how information redundancy and answer aggregation contribute to the observed sharing patterns.

2 Related Work

Communication and Evidence Routing in Multi-Agent LLMs. Recent work reduces communication overhead in multi-agent LLM systems through message pruning, adaptive communication links, and agent-specific context selection (Zhang et al., 2025; 2024; Lu et al., 2026; Liu et al., 2025; Sun et al., 2024; Guan et al., 2026; Hu et al., 2026). Complementary approaches use shared memory or context stores to control access to previously discovered information (Mao and Mirhoseini, 2026; Yu et al., 2026; Hao et al., 2026; Cuadros et al., 2026; Wu et al., 2026), or maintain separate public and private information views (Li et al., 2026). These methods address when, where, or what information should be communicated. Our focus is narrower: given a piece of discovered evidence, we study how the value of sharing it with one additional agent changes as the same evidence becomes available to more members of the team. This problem is also distinct from task allocation and agent selection

(Yue et al., 2025; Ruan et al., 2026; Hosseini et al., 2026), and from cooperative retrieval methods that coordinate what agents search for (Song et al., 2026).

Counterfactual Credit Assignment and Difference Rewards. Multi-agent reinforcement learning has long studied how to assign team-level outcomes to individual actions. Difference rewards and counterfactual credit assignment estimate an action's contribution by comparing team outcomes under closely matched alternatives (Wolpert and Tumer, 2001; Foerster et al., 2018). Related work on learned communication jointly optimizes what agents communicate, whom they communicate with, or when communication occurs (Sukhbaatar et al., 2016; Das et al., 2019; Singh et al., 2019). Our setting differs in that the LLM agents and their communication schedule are otherwise fixed. MERGE uses paired interventions that start from the same intermediate team state and differ only in whether one additional agent receives the evidence, providing a controlled training signal for how that routing decision changes final team performance.

Information Redundancy and Collective Decision-Making. Additional information can be redundant in some settings and complementary in others (Krause and Guestrin, 2005). Work on collective judgment and information cascades similarly shows that correlation among information sources can change the benefit of combining multiple views (Pivato, 2017; Bikhchandani et al., 2024). Recent studies of multi-agent LLMs report related phenomena, including reduced diversity, early convergence, and conformity (Chen et al., 2026; Cui et al., 2026; Zhu et al., 2026), while multi-agent debate and self-consistency exploit multiple reasoning paths to improve final predictions (Du et al., 2023; Wang et al., 2023). Our work connects these observations to evidence routing by studying how the benefit of broader sharing varies with the number of agents that already have the evidence. In our fixed-team setting, this variation reflects both information redundancy and the way the final aggregation rule converts individual agent outputs into a team decision.

3 PROBLEM FORMULATION

We consider a fixed team of N agents that search for evidence over multiple rounds and communicate between rounds. The retrieval procedure and final answer aggregation rule are fixed. At each communication step t, the routing problem is to decide which discovered evidence should be shared with which agents.

A routing action A specifies a set of pairs (x, i), where x is a discovered evidence item and (x, i) means that item x is routed to agent i. The same evidence item may be routed to more than one agent. Importantly, these routing decisions need not be independent: the value of routing evidence to one agent may change after the same information has already been made available to other agents.

For example, consider a four-agent fact-verification task. Suppose Agent 1 finds a useful piece of evidence. Sharing it with Agent 2 may improve the team's prediction, while sharing the same information with every remaining agent may add little. In another case, informing one more agent may change the team's final decision. The routing problem therefore requires deciding not only whether evidence is useful to an individual agent, but also how broadly it should be shared.

The next section introduces MERGE, which models this dependence and uses it together with agent-specific usefulness and communication cost to select routing actions jointly.

4 MERGE: GROUP-AWARE EVIDENCE ROUTING

Using the item-level routing decisions defined in Section 3, MERGE adds two components. First, it models how the benefit of sharing the same information changes as more agents have access to it. Second, it evaluates routing decisions jointly rather than independently. Figure 2 illustrates the overall pipeline.

Tracking Shared Information. Different evidence items may convey the same underlying information. To recognize such cases, MERGE groups semantically similar items into clusters. Let c denote one such cluster. Evidence items are still routed individually as defined in Section 3; clustering is used only to track how widely the same information is already available across the team.

For cluster c, let h denote its current number of holders, i.e., the number of agents that have access to at least one item in the cluster. The holder count allows MERGE to distinguish between evidence currently known by only one agent and evidence that is already available to most of the team.

Routing Objective. For a routing action A, let

$$A_i = \{x : (x, i) \in A\}$$

denote the evidence routed to agent i. MERGE scores the action as

$$F_t(A) = \sum_{i=1}^{N} U_{\theta,i,t}(A_i) + w_g \sum_{c} G_\phi(c, A) - C_t(A), \tag{1}$$

where $U_{\theta,i,t}(A_i)$ measures how useful the routed evidence is to agent i, $G_\phi(c, A)$ captures how the value of sharing cluster c changes with its number of holders, and $C_t(A)$ accounts for communication cost.

The agent-specific utility $U_{\theta,i,t}$ is computed by a frozen local scorer. It uses a text encoder and a lightweight MLP to estimate how well candidate evidence addresses the agent's unresolved information needs. The communication cost combines added context load, transmitted tokens, and multiple items from the same cluster routed to the same agent:

$$C_t(A) = w_{\text{load}} C_{\text{load}}(A) + w_{\text{tok}} T(A) + w_{\text{dup}} C_{\text{dup}}(A). \tag{2}$$

Detailed definitions of the local scorer and the three cost terms are given in Appendix A.

4.1 LEARNING THE BENEFIT OF ONE ADDITIONAL AGENT

To learn the group effect, we compare the final team outcome with and without one additional routing decision.

Consider a cluster c currently held by h agents and a routing decision (x, i) with $x \in c$, where agent i does not yet have evidence from the cluster. Starting from the same intermediate team state, we run two matched continuations: one includes (x, i) and the other does not. The downstream communication schedule is generated once by a frozen continuation policy π_{cont} and replayed in both continuations. Cluster c is not routed to agent i again later, so the comparison remains between receiving the evidence at this step and not receiving it. Both continuations run to the final answer, allowing the comparison to include resulting changes in later queries, retrieved evidence, and agent answers (Appendix A.4).

The predictor g_ϕ is trained to estimate the resulting difference in final team F1:

$$g_\phi \approx \mathbb{E}\left[\mathrm{F1}_T^{\text{with } (x,i)} - \mathrm{F1}_T^{\text{without } (x,i)}\right]. \tag{3}$$

The benefit of sharing with one more agent can depend on when the routing decision is made, how many agents already have the information, the team size, the position of the holder count relative to the majority threshold, and the relevance of the evidence. We encode the search stage as

$$d_t = \frac{T_{\text{comm}} - t}{T_{\text{comm}} - 1}, \tag{4}$$

for $T_{\text{comm}} > 1$, with $d_t = 0$ when there is only one communication step. Larger d_t corresponds to an earlier communication step.

Because the fixed aggregation rule used in our experiments is based on majority voting, we also define

$$\delta = \left(\left\lfloor \frac{N}{2} \right\rfloor + 1\right) - h, \tag{5}$$

which measures the distance from the current holder count to the majority threshold.

Finally, $q_c \in [0, 1]$ represents the relevance of cluster c. It is the maximum normalized support score among the items in the cluster, computed by the frozen local scorer. Unlike $U_{\theta,i,t}$, which evaluates evidence for a particular agent, q_c is a cluster-level feature used only by the group predictor.

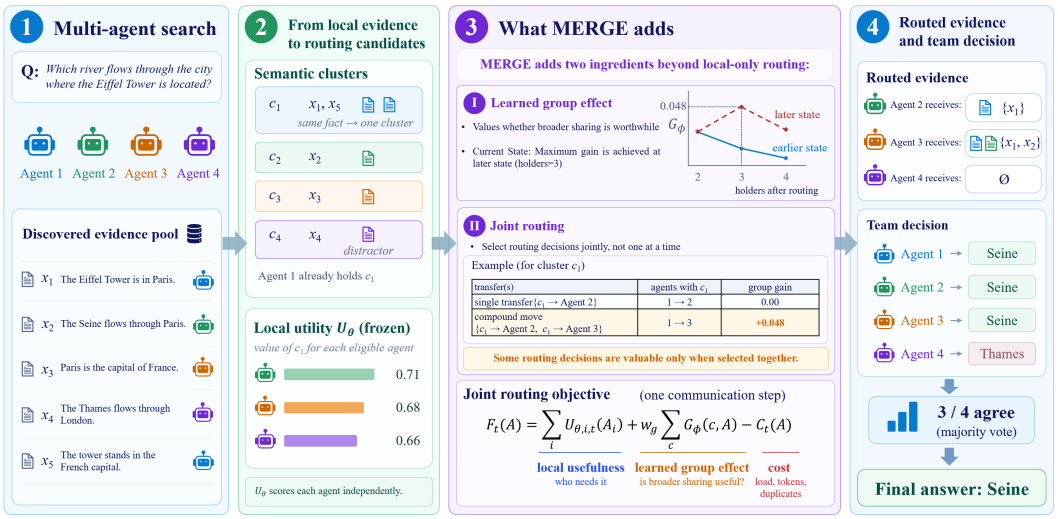


Figure 2: **Overview of MERGE.** Retrieved evidence is grouped into semantic clusters to track shared information. An agent-specific scorer estimates how useful the evidence is to each agent. MERGE learns how the benefit of sharing changes as the same information reaches more agents, and uses this group effect to select routing decisions jointly before the fixed team aggregation step.

The predictor receives

$$\left(d_t, \frac{h}{N}, \frac{\delta}{N}, \delta, \frac{1}{N}, q_c\right)$$

as input and is implemented as a one-hidden-layer MLP with 16 $\tanh$ hidden units. For compactness, we write $g_\phi(t, c, h)$ for its prediction at step t for cluster c with h current holders; the notation implicitly includes the remaining inputs above. Architecture and training details are given in Appendix A.5.

4.2 Constructing the Group Effect

Since g_ϕ is trained on differences in final team F1, its prediction can include agent-specific usefulness already modeled by U_θ. Every discovered cluster has at least one holder, namely the agent that retrieved the evidence. Thus, $h = 1$ is the smallest possible holder count and provides a natural reference. We retain only how the predicted benefit changes relative to this reference as the same information reaches more agents.

Suppose cluster c has h holders before routing action A and h' holders after the action. We define its group-effect contribution as

$$G_\phi(c, A) = \sum_{r=h}^{h'-1} \left[g_\phi(t, c, r) - g_\phi(t, c, 1)\right]. \tag{6}$$

The subtraction measures each predicted benefit relative to the one-holder case, while the sum accounts for each additional holder created by the routing action. If the action does not increase the number of holders, then $G_\phi(c, A) = 0$.

This construction makes routing decisions for the same information depend on one another. Once one additional agent receives evidence from cluster c, the holder count changes, which can change the predicted benefit of routing the same information to another agent. A local-only router does not capture this dependence because it evaluates each agent separately.

4.3 JOINT ROUTING

Because the value of one routing decision can change after another agent receives the same information, scoring routing decisions one at a time can miss useful combinations. MERGE therefore evaluates routing actions jointly.

The search maintains a bounded beam of routing actions. Starting from these actions, it considers adding, removing, or swapping one routing decision, as well as changing two routing decisions together. A change is retained only if it improves $F_t(A)$. The search stops when no candidate change improves the score or when the preset move limit is reached (Appendix A.5.4).

All controlled variants use the same search procedure; the search ablation changes only this procedure while keeping the routing objective fixed.

5 EXPERIMENTAL SETUP

We evaluate whether the two components introduced above—modeling the group effect and selecting routing decisions jointly—improve evidence routing, and whether the learned router transfers beyond its training setting. Our evaluation addresses four questions:

RQ1 (Routing Quality): Does modeling the group effect improve answer quality and communication efficiency?

RQ2 (Prediction): Can g_ϕ predict the benefit of sharing evidence with one additional agent on held-out examples?

RQ3 (Generalization): Does the learned router transfer across datasets, tasks, LLM backbones, and larger team sizes without refitting?

RQ4 (Joint Routing): Does joint routing outperform greedy routing?

Environment and Evaluation Protocol. We evaluate on three multi-hop question answering benchmarks: HotpotQA, MuSiQue, and 2WikiMultiHopQA (2WikiMHQA) (Yang et al., 2018; Trivedi et al., 2022; Ho et al., 2020), using Qwen2.5-14B, Qwen3-14B, and Gemma-3-12B (Yang et al., 2025b;a; Gemma Team, 2025). We additionally evaluate zero-shot transfer to HoVer fact verification (Jiang et al., 2020) without refitting; details are given in Appendix B.13.

Unless otherwise stated, the team contains four agents. Each agent runs four search rounds and retrieves five previously unseen documents per query using BM25 (Lin et al., 2021).

Final answers use strict majority voting. If no strict majority exists, we apply plurality followed by fixed source-count and lexical tie-breakers. This aggregation rule is fixed throughout the main evaluation.

Each dataset–backbone setting contains 400 evaluation questions with three decoding repetitions per question, matched across compared methods. The primary HotpotQA/Qwen2.5-14B setting is also used for the cost and ablation analyses.

Training and Transfer. The agent-specific utility model U_θ and group predictor g_ϕ are trained only on four-agent HotpotQA/Qwen2.5-14B development data and are frozen before evaluation. The two components use separate training records. For g_ϕ, model fitting uses 1,000 questions, and model and hyperparameter selection use a disjoint set of 250 questions. The resulting paired-intervention sets contain 9,000 fitting pairs and 2,300 selection pairs. Full training details for U_θ and g_ϕ are given in Appendix A.

For transfer experiments, the trained models are reused without refitting across QA datasets, LLM backbones, HoVer, and larger team sizes.

The communication-cost coefficients in $C_t(A)$ are selected once on the primary development data and then frozen across routing methods. The group-effect weight w_g is selected on the held-out selection split. Implementation and hyperparameter details are reported in Appendix A.

Baselines. We compare MERGE with published routing methods, controlled variants, and architecture-level reference points.

FULL-COMM broadcasts all available evidence to every agent. Among the controlled variants, SEPARABLE removes the group-effect term while retaining the same candidate evidence, agent-specific utility, communication cost, and joint routing procedure. FIXED-SHAPE replaces the learned group effect with a fixed-form alternative whose stage-dependent weighting is learned, while keeping the remaining routing components unchanged.

We also adapt RCR-Router, AgentPrune, and DYTOPO-STYLE to the same search environment; Appendix D.2 describes these adaptations. Finally, NO-COMM, RANDOM, and SINGLE-AGENT provide architecture-level reference points. The single-agent reference uses four actual queries, compared with 16 for the four-agent team under the same 16-query ceiling.

Metrics and Uncertainty. The primary metric is token-level F1. Exact Match is reported in Appendix B.11, while the HoVer transfer additionally reports verdict accuracy and Evidence F1.

Paired 95% confidence intervals are computed by question-level bootstrap resampling after averaging the three matched decoding repetitions within each question. Online token and latency measurements include all model calls. Appendix D gives the full statistical protocol.

6 RESULTS

6.1 RQ1/RQ3: QUALITY ACROSS DATASETS AND BACKBONES

MERGE achieves the highest mean F1 in all nine QA dataset–backbone settings (Table 1), with an average margin of 1.46 points over the strongest comparator. In the primary HotpotQA/Qwen2.5-14B setting, MERGE reaches 0.716 F1 versus 0.692 for FIXED-SHAPE, a 2.4-point gain with paired 95% CI $[+1.2, +3.3]$. It also leads all nine settings in mean Exact Match (Appendix B.11); strongest-comparator margins are reported in Table 9.

Margins over the strongest comparator range from 0.5 to 2.5 F1 points: they are largest on HotpotQA with Qwen2.5 and Qwen3 and smaller for Gemma on MuSiQue and 2WikiMHQA. On three additional HotpotQA backbones, MERGE improves by 1.2–2.2 points (Appendix B.1). On HoVer, the frozen router reaches 0.744 accuracy and 0.641 Evidence F1, exceeding FIXED-SHAPE by 1.3 and 1.9 points (Appendix B.13). The three primary-setting fits obtain 0.712–0.719 F1 (mean 0.716; sample SD 0.004; Table 7). All nine margins are positive.

6.2 RQ1: ONLINE QUALITY–COST TRADE-OFF

Relative to FIXED-SHAPE, MERGE gains 2.4 F1 points while saving online 1,506 tokens (11.8%) and 3.3 seconds (10.7%) per question (Table 2). Compared with FULL-COMM, it gains 5.3 points while reducing tokens by 38.9% and latency by 30.5%. DYTOPO-STYLE is cheaper and faster, but trails MERGE by 3.0 F1 points.

Against SEPARABLE, which fixes candidate evidence, local scoring, costs, and the solver, MERGE gains 3.6 F1 points while using 1,683 fewer tokens and 4.3 fewer seconds. Against DYTOPO-STYLE, it trades 287 tokens and 1.8 seconds for 3.0 F1 points, a different operating point rather than strict dominance.

A four-trajectory single-agent self-consistency reference reaches 0.629 F1, versus 0.716 for MERGE under comparable query budgets (Table 10). These resource comparisons cover online calls. Offline collection requires on the order of 10^5 matched continuations; Appendix D.3 reports the workload and amortized accounting.

6.3 RQ1/RQ4: GROUP EFFECTS AND JOINT ROUTING

Removing the group effect reduces F1 by 3.6 points, while replacing joint routing with greedy construction loses 3.0 points (Table 3). The 2×2 comparison shows why both components matter: adding group effects gains 1.1 points with greedy routing but 3.6 with joint routing, a $+2.5$-point

Table 1: Answer F1 across three backbones and three datasets. H: HotpotQA; M: MuSiQue; W: 2WikiMHQA. Small annotations indicate relative change from NO-COMM.

	Qwen2.5-14B			Qwen3-14B			Gemma-3-12B		
Method	**H**	**M**	**W**	**H**	**M**	**W**	**H**	**M**	**W**
Reference points									
NO-COMM	0.622	0.345	0.610	0.674	0.388	0.663	0.609	0.337	0.601
FULL-COMM	0.663↑6.6%	0.378↑9.6%	0.650↑6.6%	0.706↑4.7%	0.420↑8.2%	0.697↑5.1%	0.642↑5.4%	0.369↑9.5%	0.627↑4.3%
RANDOM	0.654↑5.1%	0.350↑1.4%	0.635↑4.1%	0.690↑2.4%	0.408↑5.2%	0.672↑1.4%	0.625↑2.6%	0.343↑1.8%	0.617↑2.7%
SINGLE-AGENT	0.612↓1.6%	0.337↓2.3%	0.604↓1.0%	0.655↓2.8%	0.379↓2.3%	0.649↓2.1%	0.596↓2.1%	0.326↓3.3%	0.592↓1.5%
Published routing (adapted)									
RCR-Router	0.668↑7.4%	0.382↑10.7%	0.652↑6.9%	0.710↑5.3%	0.425↑9.5%	0.702↑5.9%	0.649↑6.6%	0.374↑11.0%	0.636↑5.8%
AgentPrune	0.679↑9.2%	0.382↑10.7%	0.675↑10.7%	0.720↑6.8%	0.438↑12.9%	0.696↑5.0%	0.636↑4.4%	0.379↑12.5%	0.641↑6.7%
DYTOPO-STYLE	0.686↑10.3%	0.388↑12.5%	0.677↑11.0%	0.721↑7.0%	0.435↑12.1%	0.699↑5.4%	0.641↑5.3%	0.385↑14.2%	0.650↑8.2%
Controlled comparators									
SEPARABLE	0.680↑9.3%	0.386↑11.9%	0.664↑8.9%	0.721↑7.0%	0.434↑11.9%	0.704↑6.2%	0.653↑7.2%	0.386↑14.5%	0.653↑8.7%
FIXED-SHAPE	0.692↑11.3%	0.391↑13.3%	0.678↑11.1%	0.732↑8.6%	0.431↑11.1%	0.708↑6.8%	0.659↑8.2%	0.385↑14.2%	0.655↑9.0%
Ours									
MERGE	**0.716**↑15.1%	**0.402**↑16.5%	**0.690**↑13.1%	**0.757**↑12.3%	**0.456**↑17.5%	**0.720**↑8.6%	**0.677**↑11.2%	**0.391**↑16.0%	**0.661**↑10.0%

Table 2: Primary HotpotQA/Qwen2.5-14B results (400 questions). Δ and paired 95% CIs are MERGE minus the row, in F1 points. Tokens and time cover all online calls; token ratios are relative to FULL-COMM.

Method	**F1**	**Δ**	**Paired 95% CI**	**Tokens**	**Token ratio**	**Time (s)**
Reference points						
FULL-COMM	0.663	+5.3	$[+4.1, +6.7]$	18,462	1.000×	39.7
Routing methods and controlled variants						
SEPARABLE	0.680	+3.6	$[+2.4, +4.9]$	12,957	0.702×	31.9
RCR-Router	0.668	+4.8	$[+3.8, +6.0]$	14,419	0.781×	33.8
AgentPrune	0.679	+3.7	$[+2.2, +5.0]$	13,614	0.737×	36.2
DYTOPO-STYLE	0.686	+3.0	$[+1.8, +4.7]$	10,987	0.595×	25.8
FIXED-SHAPE	0.692	+2.4	$[+1.2, +3.3]$	12,780	0.692×	30.9
MERGE	**0.716**	—	—	11,274	0.611×	27.6

interaction (95% CI $[+1.3, +4.2]$; Appendix B.9). Constraining predicted benefits to be nonincreasing loses 1.4 points.

The nine-cell lookup table reaches 0.699 F1, recovering roughly half of the full improvement over SEPARABLE and finishing 1.7 points below the neural predictor. This small, inspectable model shows that much of the group effect can be described by a low-dimensional state. Removing the remaining-step input or one-holder reference loses 1.8 and 2.1 points; raw (h, N) encoding loses only 0.4 points.

Table 3: Primary-setting ablations and group-effect variants. Δ is variant minus full MERGE, in F1 points.

Variant	**F1**	**Δ (pp)**
Full MERGE	0.716	—
No group effect (SEPARABLE)	0.680	−3.6
Raw count/team-size encoding	0.712	−0.4
No remaining-step input d_t	0.698	−1.8
Nonincreasing predicted benefits	0.702	−1.4
No one-holder reference subtraction	0.695	−2.1
Greedy construction / single moves	0.686	−3.0
Matched parameter count and lightweight implementation		
FIXED-SHAPE (larger stage gate)	0.695	−2.1
Lookup-table predictions (stage, δ)	0.699	−1.7

Trace-level examples show the same mechanisms: group effects can favor complementary evidence over a redundant copy, while joint routing can select two individually unattractive routing decisions that together overturn a wrong majority (Appendix C.3).

6.4 RQ2: Predicting the Benefit of One Additional Agent

At $N = 4$, held-out MAE is 3.48 F1 points versus 4.18 for a constant predictor, Spearman correlation is 0.43, and all metrics weaken with team size (Table 19). End-to-end evidence includes a 2.4-point loss under the state-shuffle control. The constant-sharing control trails MERGE by 2.7, 2.2, and 1.9 points at $N = 4, 6, 8$.

The profiles reveal redundancy and an aggregation effect. The early $\delta = 2 \to 1$ decline persists under individual-agent F1, whereas the sharp late peak largely vanishes, tying it to the fixed majority rule (Tables 20 and 17). Continuation-policy sensitivity appears in Table 13.

6.5 RQ3: Transfer to Larger Teams

Without refitting, MERGE leads at $N = 4, 6, 8$ under the same evaluation protocol, beating SEPARABLE by 3.6, 3.5, and 3.6 F1 points and FIXED-SHAPE by 2.4, 2.0, and 1.8 (Tables 4 and 8). F1 peaks at six agents; consistent gains support transfer despite weaker calibration.

Table 4: Answer F1 by team size on HotpotQA/Qwen2.5-14B without refitting the four-agent model.

Method	$N = 4$	$N = 6$	$N = 8$
Reference points			
FULL-COMM	0.663	0.668	0.654
Routing methods and controlled variants			
SEPARABLE	0.680	0.687	0.679
RCR-Router	0.668	0.675	0.674
AgentPrune	0.679	0.686	0.688
DYTOPO-STYLE	0.686	0.694	0.690
FIXED-SHAPE	0.692	0.702	0.697
MERGE	0.716	0.722	0.715

7 Discussion and Limitations

Early diminishing returns reflect information redundancy, whereas one more late-stage holder can change the majority. Routing thus depends on both the team information state and its decision rule.

Because paired interventions freeze π_{cont} and block later delivery to the selected agent, training may differ from learned-router trajectories. The predictor also omits agent identity and cross-cluster interactions.

Limits include Wikipedia retrieval, homogeneous agents, BM25, fixed voting, out-of-distribution one-holder references at $N = 6, 8$, and approximate routing search; transfer beyond these settings remains open.

8 Conclusion

MERGE learns holder-dependent sharing value for joint routing. Across nine QA settings, it improves quality, reduces primary-setting communication, and transfers to larger teams and fact verification.

Reproducibility Statement

Code is available at `https://anonymous.4open.science/r/merge-routing-053E/`. Appendix A provides implementation and training details, including the hyperparameters in Table 5. Appendix A.5 documents the data partitions, paired-intervention procedure, fitting objective, and joint-routing search procedure. Appendix D specifies the statistical protocol, while Appendix D.3 reports sample accounting, exclusions, and offline collection costs.

Ethics Statement

This study uses established question-answering benchmarks and does not involve human subjects. The main foreseeable risks are the computational cost of multi-agent inference, propagation of errors or biases from retrieved sources, and overstatement of conclusions from noisy empirical comparisons. We mitigate these risks by tracking evidence provenance, using shared evaluation protocols, reporting resource use and uncertainty, and stating the limits of transfer beyond the studied tasks and team configurations. These measures do not eliminate the risks. Applying the method to private or sensitive documents would additionally require appropriate access controls, privacy and legal review, and mechanisms for correcting misinformation after it has been shared among agents.

AI Use Statement

In this work, we used generative AI tools only for language polishing and grammar queries. We did not use generative AI tools to formulate the research questions or methodology, conduct the literature review, generate citations, write experimental code, run experiments, analyze results, or produce research claims or artifacts. The authors reviewed every AI-assisted language edit and verified that it did not alter the intended technical meaning, evidence, or attribution. We take responsibility for the final content of this work, including all text, claims, results, citations, and artifacts.

References

Sushil Bikhchandani, David Hirshleifer, Omer Tamuz, and Ivo Welch. Information cascades and social learning. *Journal of Economic Literature*, 62(3):1040–1093, 2024.

Nuo Chen, Yicheng Tong, Yuzhe Yang, Yufei He, Xueyi Zhang, Zou Qingyun, Qian Wang, and Bingsheng He. Diversity collapse in multi-agent llm systems: Structural coupling and collective failure in open-ended idea generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 251–306, 2026. doi: 10.18653/v1/2026.findings-acl.13.

Diego F. Cuadros, Abdoul-Aziz Maiga, Helen Meskhidze, and Andre Curtis-Trudel. Governed collaborative memory as artificial selection in llm-based multi-agent systems. *arXiv preprint arXiv:2605.04264*, 2026.

Yu Cui, Hang Fu, Haibin Zhang, Licheng Wang, and Cong Zuo. Free-mad: Consensus-free multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 31977–31997, 2026. doi: 10.18653/v1/2026.findings-acl.1600.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 1538–1546. PMLR, 2019.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Yao Guan, Lin Wang, Zhihu Lu, Ziyi Wang, Wenzhu Yan, and Qiang Duan. Moc: Multi-order communication in llm-based multi-agent systems. *arXiv preprint arXiv:2606.02359*, 2026.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*, 2024.

Guangya Hao, Yunbo Long, and Zhuokai Zhao. Self-evolving multi-agent systems via decentralized memory. *arXiv preprint arXiv:2605.22721*, 2026.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of COLING*, pages 6609–6625, 2020.

Mohammad Parsa Hosseini, Ankit Shah, Saiyra Qureshi, Alex Huang, Connie Miao, and Wei Wei. Training-free agentic ai: Probabilistic control and coordination in multi-agent llm systems. *arXiv preprint arXiv:2603.13256*, 2026.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Tong Zhao, Xiaoxi Li, Zheng Liu, and Zhicheng Dou. Agentfugue: Agent scaling for long-horizon tasks through collective reasoning. *arXiv preprint arXiv:2605.24486*, 2026.

Yichen Jiang, Shikha Bordia, Zheng Zhong, Charles Dognin, Maneesh Singh, and Mohit Bansal. HoVer: A dataset for many-hop fact extraction and claim verification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3441–3460. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.309.

Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 324–331, 2005. URL `https://las.inf.ethz.ch/files/krause05near.pdf`.

Yuante Li, Yicheng Tao, Kate Zhang, Taozhi Wang, Gefei Gu, and Yaxin Zhou. Diverse evidence, better forecasts: Multi-agent deliberation under information asymmetry. *arXiv preprint arXiv:2607.01661*, 2026.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of SIGIR*, 2021.

Jun Liu, Zhenglun Kong, Changdi Yang, Fan Yang, Tianqi Li, Peiyan Dong, Joannah Nanjekye, Hao Tang, Geng Yuan, Wei Niu, et al. Rcr-router: Efficient role-aware context routing for multi-agent llm systems with structured memory. *arXiv preprint arXiv:2508.04903*, 2025.

Yuxing Lu, Yucheng Hu, Xukai Zhao, and Jiuxin Cao. Dytopo: Dynamic topology routing for multi-agent reasoning via semantic matching. *arXiv preprint arXiv:2602.06039*, 2026.

Yuzhen Mao and Azalia Mirhoseini. Decentralized multi-agent systems with shared context. *arXiv preprint arXiv:2606.10662*, 2026.

Marcus Pivato. Epistemic democracy with correlated voters. *Journal of Mathematical Economics*, 72:51–69, 2017.

Jianhao Ruan, Zhihao Xu, Yiran Peng, Fashen Ren, Zhaoyang Yu, Xinbing Liang, Jinyu Xiang, Yongru Chen, Bang Liu, Chenglin Wu, Yuyu Luo, and Jiayi Zhang. Aorchestra: Automating sub-agent creation for agentic orchestration. *arXiv preprint arXiv:2602.03786*, 2026.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*, 2019.

Lichang Song, Ting Long, and Yi Chang. Rethinking retrieval-augmented generation as a cooperative decision-making problem. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 43445–43456, 2026. doi: 10.18653/v1/2026.findings-acl.2157.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Chuxiong Sun, Zehua Zang, Jiabao Li, Jiangmeng Li, Xiao Xu, Rui Wang, and Changwen Zheng. T2mac: Targeted and trusted multi-agent communication through selective engagement and evidence-driven integration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15154–15163, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

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

Chuanjie Wu, Zhishang Xiang, Yunbo Tang, Zerui Chen, Qinggang Zhang, and Jinsong Su. Memgraphrag: Memory-based multi-agent system for graph retrieval-augmented generation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026. URL https://arxiv.org/abs/2606.00610.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, 2018.

Zhongming Yu, Naicheng Yu, Hejia Zhang, Wentao Ni, Mingrui Yin, Jiaying Yang, Yujie Zhao, and Jishen Zhao. Multi-agent memory from a computer architecture perspective: Visions and challenges ahead. *arXiv preprint arXiv:2603.10062*, 2026.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems. In *Proceedings of ACL*, pages 15549–15572, 2025.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *International Conference on Learning Representations*, volume 2025, pages 75389–75428, 2025.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 33909–33930, 2026. doi: 10.18653/v1/2026.findings-acl.1694.

APPENDIX CONTENTS

A Implementation and Reproducibility 14
A.1 Evidence state and semantic clusters 14
A.2 Agent-specific utility 14
A.3 Shared routing costs 15
A.4 Continuation policy and schedule replay 15
A.5 Learning the group predictor and joint routing 17
B Core Supplementary Experiments 19
B.1 Additional backbones 19
B.2 Training seeds and model variants 19
B.3 Larger-team paired contrasts 19
B.4 Strongest-comparator contrasts across the grid 20
B.5 Online self-consistency comparison 20
B.6 Constant sharing-increment control 21
B.7 Fixed-composition role-specialized pilot 21
B.8 Continuation-policy sensitivity 21
B.9 Group effects and joint routing 22
B.10 Matching predicted benefits to the state 22
B.11 Exact Match robustness evaluation 23
B.12 Aggregation-free outcome sensitivity 23
B.13 Transfer to fact verification 24
C Understanding Group Effects and Routing Decisions 25
C.1 Predicting the benefit of one additional agent 25
C.2 Measured and predicted holder-count patterns 26
C.3 Trace-level case studies 27
D Statistical Protocol and Baseline Adaptations 31
D.1 Statistical protocol and scope 31
D.2 Adapted published routers 31
D.3 Sample counts, exclusions, and cost accounting 31

A Implementation and Reproducibility

This appendix explains how we build evidence items, score their usefulness to each agent, and account for communication costs. It also defines the policy used to continue paired interventions. Appendix A.5 describes how we learn the benefit of sharing with one additional agent and perform joint routing.

A.1 Evidence state and semantic clusters

Each retrieved document is split into short, source-backed evidence items. An item stores its text, source title, document identifier, token length, named entities, discoverer, the agents that already have it, and retrieval round. Items from different sources remain separate. Items with similar meaning can share a cluster identifier, but this does not merge their text or source records. The router can therefore recognize repeated evidence without losing where it came from.

Clusters are built online with a frozen sentence encoder. A new item joins the closest existing cluster when cosine similarity exceeds 0.85, selected on development data; otherwise it starts a new cluster. The clustering code does not use gold supporting facts or final answers. Table 5 reports the selected clustering threshold alongside the scorer and routing hyperparameters.

For agent i, the state contains its existing items and clusters, cumulative received tokens, recent queries, retrieved titles, and unresolved evidence requirements. A requirement is a short description of missing information, produced from the question and the agent's current trace. Malformed requirements are ignored rather than used to exclude candidate evidence items.

A.2 Agent-specific utility

Let $\mathcal{G}_i$ be agent i's unresolved requirements and s_i its current state. The agent-specific utility is

$$U_{\theta,i,t}(A_i) = \sum_{g\in\mathcal{G}_i} w_{ig}\left[1-\exp\left(-\sum_{x\in A_i} r_\theta(x,g,s_i)\right)\right] + \sum_{x\in A_i}\widetilde{v}_{\theta,i,t}(x) + U^{\mathrm{corr}}_{i,t}(A_i), \tag{7}$$

where $r_\theta(x,g,s_i)\geq 0$ estimates how well item x supports requirement g, and $\widetilde{v}_{\theta,i,t}(x)$ adjusts the item's score relative to other candidates for that agent. Each valid requirement has weight $w_{ig}=1$, without normalization across requirements. The coverage term is therefore bounded by $|\mathcal{G}_i|$, rather than a fixed unit total; it is zero when no valid requirements remain. Additional evidence for an already covered requirement still has diminishing marginal value, while covering another requirement adds its own utility. A frozen text encoder represents the evidence item, requirement, question, and agent's last query. Their embeddings are combined with scalar state features: current load, holder count, retrieval round, source support, and whether the item introduces a new entity. A two-layer MLP predicts r_θ through a softplus output and predicts the raw residual v_θ through $v_{\max}\tanh(\cdot)$. Numerical settings are consolidated in Table 5.

Subtracting the mean item adjustment. Let $\mathcal{X}^{\mathrm{cand}}_{i,t}$ be the fixed set of candidate evidence items for agent i before routing. For a nonempty set, define

$$\overline{v}_{\theta,i,t} = \frac{1}{|\mathcal{X}^{\mathrm{cand}}_{i,t}|}\sum_{x'\in\mathcal{X}^{\mathrm{cand}}_{i,t}} v_\theta(x',s_i), \qquad \widetilde{v}_{\theta,i,t}(x) = v_\theta(x,s_i)-\overline{v}_{\theta,i,t}. \tag{8}$$

For an empty set, set the mean to zero; there are no candidate evidence items. The mean is computed once per agent and communication step, over all candidates rather than the selected evidence set or a training minibatch. It stays fixed during beam search and refinement. Subtracting this mean removes an offset shared by all candidates, while preserving differences between their adjustments. The raw bound is $|v_\theta|\leq v_{\max}$; the centered bound is $|\widetilde{v}_{\theta,i,t}|\leq 2v_{\max}$. This agent-specific mean subtraction is separate from the one-holder reference used in the group effect.

All local features are computed before routing and held fixed while candidate routing actions are compared. In particular, the local holder-count feature stays fixed when another agent is added to an action. The change in value caused by that addition is represented by G_ϕ.

The remaining correction rewards relevance to the question and penalizes overlap with evidence the agent already has:

$$U_{i,t}^{\mathrm{corr}}(A_i) = \sum_{x \in A_i} \left[\alpha_{\mathrm{question}} q_{\mathrm{question}}(x) - \alpha_{\mathrm{held}} q_{\mathrm{held}}(x, i)\right].$$

Both relevance and overlap features lie in $[0, 1]$. Requirement weights and correction coefficients are fixed across evaluation settings.

Training pairs are formed within the same agent state. A candidate evidence item is preferred when the following search round recovers a new benchmark supporting document, closes an annotated requirement, or improves the agent's answer F1. The loss combines pairwise logistic ranking with a binary prediction term predicting whether the routing decision is productive:

$$\mathcal{L} = -\log \sigma(s_+ - s_-) + \beta_{\mathrm{BCE}} \,\mathrm{BCE}(\sigma(s), y). \tag{9}$$

Here s is the raw single-item routing utility: Equation 7 for one evidence item, using v_θ in place of $\widetilde{v}_{\theta,i,t}$. The logits s_+ and s_- correspond to preferred and nonpreferred evidence items, and y is the binary productivity label. BCE uses the raw logit; routing uses the mean-subtracted item adjustment. Within one agent state, subtracting the mean leaves the difference between two single-item scores unchanged. For variable-size selected evidence sets, it removes the term $|A_i|\overline{v}_{\theta,i,t}$, so a learned residual baseline does not act as a uniform per-item penalty or reward. Requirement coverage, relative item adjustments, communication costs, and routing constraints still determine the number of selected items. Negative examples come from the same candidate pool, keeping the comparison within a common state. The scorer is trained on HotpotQA/Qwen2.5-14B development trajectories and frozen for the QA evaluation grid.

Training partitions. Fitting and model selection use disjoint question sets. All trajectories and pairs from one question remain together. Missing continuation outcomes are not treated as negative labels. Checkpoints are selected only on the selection set, never on evaluation questions or evaluation datasets. The local scorer and group predictor have separate training records.

A.3 Shared routing costs

The objective in Equation 1 uses

$$C_{\mathrm{load}}(A) = \sum_i \big[C(L_{i,t} + T_i(A)) - C(L_{i,t})\big], \qquad C(\ell) = \ell^2 / S_{\mathrm{load}}^2,$$

and

$$C_{\mathrm{dup}}(A) = \sum_{i,c} \min(\max(n_{ic}(A) - 1, 0), 2),$$

where $L_{i,t}$ is the agent's token load before the step, and $T_i(A)$ is the new load from the action. The count $n_{ic}(A)$ is the number of selected items from cluster c routed to agent i under action A. The first cost penalizes load; the second penalizes repeated evidence within the selected evidence set, with a cap for each agent–cluster pair. This is distinct from overlap with evidence the agent already has and from G_ϕ, which accounts for sharing the same evidence with different agents. Together with $T(A)$, these give $C_t(A) = w_{\mathrm{load}} C_{\mathrm{load}}(A) + w_{\mathrm{tok}} T(A) + w_{\mathrm{dup}} C_{\mathrm{dup}}(A)$. The generic communication-cost weights $w_{\mathrm{load}}, w_{\mathrm{tok}}, w_{\mathrm{dup}}, S_{\mathrm{load}}$ are selected once on development data and frozen across all routing methods. Group-effect weights are selected separately for each formulation. Table 5 lists the weights and training settings.

Cluster relevance q_c is the maximum normalized predicted support mass within the cluster, clipped to $[0, 1]$ and cached before routing. This normalization applies to the group predictor's input, not to the requirement weights in U_θ. Changing A changes how many agents have the evidence, not its assigned relevance.

A.4 Continuation policy and schedule replay

We use π_{cont} to generate logged pre-intervention states and the communication schedule used after an intervention. The schedule is generated once and then replayed in both matched continuations,

Table 5: Implementation and selected hyperparameters. Generic communication costs are selected once on development data and frozen across routing methods; group-effect weights are selected separately by formulation.

Component	Configuration
Semantic-cluster cosine threshold	0.85; selected on development data
Local scorer hidden width	256
Requirement weights	$w_{ig} = 1$ per valid requirement; no normalization
Question / held-context correction	$\alpha_{\text{question}} = 0.20$; $\alpha_{\text{held}} = 0.20$ (subtracted)
Residual centering	Per agent and step, over the fixed candidate set
Raw / centered residual bound	0.25 / 0.50
Local BCE weight / input	$\beta_{\text{BCE}} = 0.2$ / raw single-item routing logit
Load / token / duplicate weights	$w_{\text{load}} = 0.5$; $w_{\text{tok}} = 0.0005$; $w_{\text{dup}} = 0.1$
Load normalization / duplicate cap	$S_{\text{load}} = 3000$ tokens / 2 per agent–cluster
Group-effect weight: MERGE	$w_g = 30$
Group-effect weight: schedule generation	$w_g^{\text{cont}} = 0.4$; frozen during label collection
Group predictor	6 inputs; 16 tanh units; 129 parameters
Fixed-form stage gate	2 parameters
Larger fixed-form stage gate	2 inputs; 32 tanh units; 129 parameters
Lookup-table predictions	9 stage–δ cells
Optimizer / learning rate / batch size	AdamW / 0.001 / 256
Maximum / completed / selected epoch	100 / 100 / 100
L2 / smoothness weight	$\lambda = 10^{-4}$ / $\eta = 0.02$
Model seeds	0 primary; 1 and 2 stability
Selection criterion	Question-balanced held-out squared error
Fitting / selection questions	1,000 / 250
Fitting / selection pairs	9,000 / 2,300
Beam width / refinement cap	16 actions / 64 accepted moves

so the router does not re-optimize after the selected routing decision changes the state. The policy determines the communication plan, while the paired comparison holds that plan fixed. Its scorer, weights, and solver are frozen before collection. It is a data-collection policy, not an evaluated baseline. For each cluster c, let h_c denote its holder count before routing and $h_c(A)$ its holder count after routing action A. Let $m = \lfloor N/2 \rfloor + 1$ denote the majority threshold. Define

$$f_{\text{broad}}(A) = \sum_c \left[\max\big(h_c(A) - h_c - 1, 0\big)\right]^2,$$
$$f_{\text{reach}}(A) = \sum_c q_c \frac{\min(h_c(A), m) - \min(h_c, m)}{m - 1}. \tag{10}$$

The first term penalizes sharing with several additional agents; the second rewards broader sharing up to m agents. With $\tau_t = 1 - d_t$ (and $\tau_t = 1$ for a single communication step), the continuation policy scores actions by

$$F_t^{\text{cont}}(A) = \sum_i U_{\theta,i,t}(A_i) + w_g^{\text{cont}}\left[\tau_t f_{\text{reach}}(A) - (1 - \tau_t) f_{\text{broad}}(A)\right] - C_t(A). \tag{11}$$

Communication costs are shared with MERGE; the continuation policy has its own frozen group-effect weight, $w_g^{\text{cont}} = 0.4$. The saturation at m and the stage-dependent balance are hand-specified. These rules determine the communication plan that is replayed, rather than new routing decisions within each matched continuation. Appendix B.8 compares schedules generated by this policy with schedules generated by SEPARABLE, which omits both group-effect rules.

Under the same candidate set and objective components, the continuation solver starts from the empty action, greedily adds feasible positive-gain routing decisions, and then applies improving add, drop, and one-for-one swap moves. Refinement stops when no move improves the score or the configured move cap is reached. The resulting schedule is replayed in both matched continuations to the terminal answer. The selected cluster is blocked from later delivery to the selected agent. A scheduled routing decision is skipped if its agent already has the item or if it is no longer admissible

under the replayed state. Such skips do not trigger replacement routing decisions or re-optimization: the planned schedule stays fixed, although the executed routing decisions can differ under these checks. Search and final answers remain free to diverge.

The FIXED-SHAPE controlled variant uses the same two sharing functions, but learns their stage-dependent balance and uses the common joint solver described in Appendix A.5.

A.5 LEARNING THE GROUP PREDICTOR AND JOINT ROUTING

This section describes how we collect paired interventions, fit the group predictor, and perform joint routing.

A.5.1 DATA PARTITIONS AND PAIRED INTERVENTIONS

Four disjoint question sets serve different purposes. Discovery examines how the benefit changes with holder count, fitting trains g_ϕ, selection chooses regularization and routing weights, and confirmation evaluates the frozen router. These sets are also separate from local-scorer training. All states, repetitions, clusters, and matched continuations from one question stay in the same set.

Logged intervention states come from trajectories of the frozen collection policy π_{cont} (Appendix A.4). For an eligible cluster c and target holder count h, we construct a feasible routing context in which c is available to h agents while excluding agent i. We then compare two matched continuations that differ only in whether an item $x \in c$ is routed to i. The routing context, selected item, item length, and pre-intervention agent state and local scores are shared.

The remaining communication schedule is generated once and replayed in both matched continuations with matched decoding repetitions. Cluster c is not delivered later to the selected agent, so the comparison does not collapse into earlier versus later delivery. A replayed routing decision is skipped if the agent already has the item or if the decision is no longer feasible under the replayed state. Subsequent queries, retrieved evidence, and final answers can differ. The training target is the difference in final team F1 between the two matched continuations under this fixed plan, rather than next-step F1 or evidence recall.

Within each eligible state, clusters are sampled before observing outcomes, with probability proportional to $1/(r+1)$, where r is their rank by pre-routing agent-specific utility, starting at zero for the highest rank. Agents are sampled by a balanced rule, also before outcomes are observed. Neither choice depends on the observed intervention effect. Cluster relevance is computed before the intervention and held fixed.

With four search rounds, early, middle, and late communication have $d_t = 1, 1/2, 0$. For $N = 4, 6, 8$, the values $\delta = 2, 1, 0$ correspond to holder counts $(1, 2, 3)$, $(2, 3, 4)$, and $(3, 4, 5)$. A cluster contributes only target holder counts reachable from its current logged holder count: both matched conditions must admit the routing decision under the same pre-intervention state, and the selected agent must not already have cluster evidence. Neighboring comparisons use states where both target holder counts admit eligible intervention pairs. The $h = 1$ input is a mathematical reference for the learned model, not a common starting state for data collection; evaluating $g_\phi(t, c, 1)$ does not require resetting the logged cluster to one agent. Cross-size comparisons require the same questions to be eligible at each size. Incorrect answers and negative effects are kept. Missing states, infeasible actions, decoding failures, and broken matching conditions are recorded separately.

After eligibility filtering, fitting, selection, and prediction use fixed, pre-specified quotas balanced over search stage, pre-routing holder count and target holder count h, restricted to reachable cells. The retained sample sizes are 9,000 fitting pairs, 2,300 selection pairs, and 1,800 prediction pairs per team size (Table 21). The table also reports how many candidate configurations are excluded before these quotas are filled.

A.5.2 GROUP PREDICTOR AND FITTING OBJECTIVE

The group predictor has a tanh hidden layer and a tanh scalar output; Table 5 gives its dimensions. The raw target lies in $[-1, 1]$. Input normalization uses fitting-only statistics; constant features, including $1/N$ in the four-agent fit, are not divided by zero variance. Integer δ and $1/N$ retain their natural scales.

Team-size parameterization. The group predictor is fitted only at $N = 4$. For a fixed team size, h/N, δ/N, and δ describe the same holder count in different ways; their contributions are not separate effects. The inputs include both counts and proportions so that the model can be applied to larger teams. The $N = 6, 8$ intervention sets, each containing 1,800 pairs, are used only for evaluation. Prediction and end-to-end routing therefore test transfer without fitting on larger teams.

The parameterization also explains a source of transfer sensitivity. At fixed N, the coordinates obey

$$\frac{h}{N} + \frac{\delta}{N} = \frac{m}{N}, \qquad \delta = N\frac{\delta}{N}, \qquad \frac{1}{N} = \text{constant}.$$

At each team size, these holder-count features therefore vary along one degree of freedom. Four-agent training covers the relation $m/N = 3/4$; at $N = 6, 8$, it becomes $4/6$ and $5/8$. Two models can agree on all four-agent inputs yet make different predictions for larger teams. Training at $N = 4$ alone thus leaves the scale and ranking of larger-team predictions underdetermined. The held-out evaluation measures how well the fitted model transfers.

Let $\mathbf{z}_{js}$ denote the predictor input and y_{js} the observed final-F1 difference for intervention pair s from question j. For question j, with n_j intervention pairs, we use the loss

$$\mathcal{L}(\phi) = \frac{1}{|D_{\text{fit}}|}\sum_j \frac{1}{n_j}\sum_s \big(g_\phi(\mathbf{z}_{js}) - y_{js}\big)^2 + \lambda\|\phi\|_2^2 + \eta\,\mathbb{E}\big[(g_\phi(h+1) - 2g_\phi(h) + g_\phi(h-1))^2\big]. \tag{12}$$

The first term gives each question equal weight, regardless of how many pairs it contributes. The smoothness term compares adjacent holder counts represented in the data. All count-dependent features are recomputed together while stage and relevance stay fixed. The penalty discourages abrupt changes without forcing a particular predicted-benefit curve.

Architecture, training settings, and selected regularization weights appear together in Table 5.

Model seeds are separate from decoding repetitions. The protocol specifies seed 0 for the primary model and seeds 1 and 2 for stability checks; it does not choose among seeds using test scores. Table 7 reports all three seeds and separates model-seed variation from paired question-level uncertainty.

A.5.3 Controlled group-effect variants

FIXED-SHAPE fits $\tau_\alpha(d) = \sigma(a_0 + a_1 d)$, with $\alpha = (a_0, a_1)$, while retaining the fixed broad-sharing penalty and reach reward. It uses the same development interventions and selection budget. Only their stage-dependent balance is learned; the two sharing functions keep their fixed forms. This is the comparator in the main grid. It differs from π_{cont}, whose hand-specified stage gate is used to generate replay schedules in Appendix A.4.

The raw-encoding ablation replaces the four count/team-size features with (h, N) while retaining stage and relevance. The remaining-step ablation removes d_t, not the δ feature. The nonincreasing-benefit control constrains predicted benefits to a nonincreasing sequence before reference subtraction and summation. The no-reference variant sums raw group-predictor outputs and selects its group-effect weight on the same selection split. Search ablations keep the objective fixed. The ablation results are in Table 3.

Matched parameter count and lookup-table variant. The larger FIXED-SHAPE gate replaces the linear-logit stage gate with a nonlinear gate on (d_t, d_t^2), while retaining the same fixed sharing functions. It matches the group predictor's parameter count but has different inputs and a fixed-form group effect (Table 5). The lookup-table variant estimates question-balanced paired final-F1 differences in (stage, δ) cells using only the fitting split, then applies the same reference subtraction and summation. Its bin layout matches Table 20; the held-out outcomes in that table are excluded from fitting. Unobserved fitting bins use the fitting-set global mean, fixed before selection. Both versions use the common search procedure; their outcomes are reported in Table 3.

A.5.4 Beam search and compound refinement

The common solver starts with the empty action and uses a fixed-width beam. At each depth, it tries feasible one-decision additions, removes duplicate actions, and keeps the highest scores up to the

configured beam width. It also keeps the best action found at any depth, including the empty action. Positive item lengths and the routing constraints bound the number of additions. Ties are resolved deterministically, and methods use the same candidates.

Refinement tries adding or dropping one routing decision, swapping one for another, adding routing decisions for two agents from the same cluster, and dropping one routing decision while adding two. A move is accepted only if it improves the full objective and remains feasible. Refinement stops when no improving move remains or when the configured cap on accepted moves is reached.

Within one routing step, the group term depends on the number of additional holders, not on the order in which those routing decisions are considered. Across search rounds, an earlier routing decision can change later queries and states. The complete objective also contains agent-specific utility and communication costs, and the bounded search returns a feasible routing action without a global optimality guarantee.

B Core Supplementary Experiments

B.1 Additional backbones

Table 6 supplements the main grid with HotpotQA on three backbones. MERGE improves on the strongest comparator by 2.2, 2.2, and 1.2 points. That comparator is FIXED-SHAPE for Qwen3.6-27B and SEPARABLE for GPT-5.6 Sol and Claude Opus 5. Each comparison uses 400 HotpotQA questions. The table reports mean F1; paired intervals are not reported for these additional backbones.

Table 6: Additional-backbone comparisons on HotpotQA (reported sample size: 400 questions; mean answer F1). Δ is MERGE minus the strongest displayed comparator, in F1 points.

	Reference point	*Routing methods*			
Backbone	**FULL-COMM**	**SEPARABLE**	**FIXED-SHAPE**	**MERGE**	Δ
Qwen3.6-27B	0.756	0.761	0.768	0.790	+2.2
GPT-5.6 Sol	0.805	0.815	0.814	0.837	+2.2
Claude Opus 5	0.807	0.814	0.809	0.826	+1.2

B.2 Training seeds and model variants

Seeds 0, 1, and 2 share the same question partitions, labels, fitting budget, and evaluation protocol. Seed 0 is the primary reference; all three seeds contribute to the reported mean and sample SD.

Table 7: Primary-setting training-seed stability. SD measures variation across three fitted models, not question-level uncertainty.

Method	Seed 0	Seed 1	Seed 2	Mean	Sample SD
FIXED-SHAPE	0.692	0.688	0.694	0.691	0.003
MERGE	0.716	0.712	0.719	0.716	0.004

The larger FIXED-SHAPE gate matches the predictor's parameter count but retains fixed sharing functions and does not observe holder count. It reaches 0.695 F1. A nine-cell stage–δ lookup table reaches 0.699, recovering roughly half of the full improvement over SEPARABLE with an inspectable, negligible-cost representation.

B.3 Larger-team paired contrasts

The paired bootstrap resamples questions jointly across methods after averaging matched decoding repetitions. The four-agent row reuses the primary comparison; larger teams use the same frozen group predictor.

Table 8: MERGE minus controlled variant across team sizes, in F1 points.

N	Comparator	Mean difference	Paired 95% CI
4	SEPARABLE	+3.6	$[+2.4, +4.9]$
4	FIXED-SHAPE	+2.4	$[+1.2, +3.3]$
6	SEPARABLE	+3.5	$[+2.0, +4.9]$
6	FIXED-SHAPE	+2.0	$[+0.5, +3.4]$
8	SEPARABLE	+3.6	$[+1.9, +5.1]$
8	FIXED-SHAPE	+1.8	$[+0.3, +2.8]$

B.4 STRONGEST-COMPARATOR CONTRASTS ACROSS THE GRID

Table 9 reports mean contrasts against the strongest non-MERGE comparator in all nine settings. The primary contrast reuses the main routing study; the other eight are exploratory. Each setting uses 400 questions. Appendix D gives the comparison scope.

Table 9: MERGE minus each setting's strongest displayed non-MERGE comparator. Differences are in F1 points (400 questions per setting). Non-primary contrasts are exploratory.

Backbone	Dataset	Comparator	Δ
Qwen2.5-14B	HotpotQA	FIXED-SHAPE	+2.4
Qwen2.5-14B	MuSiQue	FIXED-SHAPE	+1.1
Qwen2.5-14B	2WikiMHQA	FIXED-SHAPE	+1.2
Qwen3-14B	HotpotQA	FIXED-SHAPE	+2.5
Qwen3-14B	MuSiQue	AgentPrune	+1.8
Qwen3-14B	2WikiMHQA	FIXED-SHAPE	+1.2
Gemma-3-12B	HotpotQA	FIXED-SHAPE	+1.8
Gemma-3-12B	MuSiQue	SEPARABLE	+0.5
Gemma-3-12B	2WikiMHQA	FIXED-SHAPE	+0.6

B.5 ONLINE SELF-CONSISTENCY COMPARISON

The primary-setting comparison gives a single search agent $k = 4$ independent retrieval-and-answer trajectories and aggregates their answers using the same normalization and voting rule as the team. In contrast, SINGLE-AGENT in Table 1 uses one trajectory and scores 0.612; the four-trajectory self-consistency variant scores 0.629. All trajectories, retrieval calls, and aggregation calls count toward total online token use. Table 10 reports realized online use over the same 400 questions. SC uses 9,879 tokens and 22.9 seconds per question, compared with 11,274 tokens and 27.6 seconds for MERGE. Thus MERGE uses 1,395 more tokens (14.1% relative to SC) and 4.7 more seconds per question. One-time training and intervention collection, including their amortized deployment cost, are reported in Appendix D.3.

Table 10: Online self-consistency comparison on HotpotQA/Qwen2.5-14B (400 questions). Tokens and time give realized online use. Δ and paired 95% CIs are MERGE minus the row, in F1 points.

System	F1	Tokens	Time (s)	Δ	Paired 95% CI
Fixed team + MERGE	0.716	11,274	27.6	—	—
Single-agent self-consistency ($k = 4$ trajectories)	0.629	9,879	22.9	+8.7	$[+7.0, +10.4]$

MERGE improves mean F1 by 8.7 points (paired 95% CI $[+7.0, +10.4]$) at the reported increase in online tokens and latency. At these resource levels, independent single-agent sampling does not recover the answer quality achieved by routing evidence within the team.

B.6 CONSTANT SHARING-INCREMENT CONTROL

This control replaces the holder-dependent relative prediction $g_\phi(t, c, r) - g_\phi(t, c, 1)$ with a single fitted constant $\bar{b}$, independent of stage, relevance, and team size. For a cluster c with h holders before routing action A and h' holders afterward, define

$$G_{\text{const}}(c, A) = \sum_{r=h}^{h'-1} \bar{b}\, \mathbf{1}\{r > 1\}.$$

We fit $\bar{b}$ to the four-agent training targets after reference subtraction and choose its weight on the four-agent selection set, then freeze both for $N = 4, 6, 8$. All other routing components are unchanged. The comparison asks whether a fixed preference for broader sharing is sufficient.

Table 11: Constant-effect control across team sizes. Δ and pointwise paired 95% CIs are MERGE minus the constant-effect control, in F1 points. The model and group-effect weights are fitted or selected only at $N = 4$.

N	SEPARABLE F1	Constant-effect F1	MERGE F1	Δ	Paired 95% CI
4	0.680	0.689	0.716	+2.7	$[+1.4, +3.7]$
6	0.687	0.700	0.722	+2.2	$[+1.0, +3.6]$
8	0.679	0.696	0.715	+1.9	$[+0.5, +3.4]$

The constant-effect control improves on SEPARABLE by 0.9, 1.3, and 1.7 points at $N = 4, 6, 8$, while MERGE adds a further 2.7, 2.2, and 1.9 points. The four-agent contrast, $[+1.4, +3.7]$, shows that the advantage of state-dependent sharing benefits is already present at the training team size. A constant effect captures part of the improvement; state-dependent sharing benefits account for the remainder, including on larger teams.

B.7 FIXED-COMPOSITION ROLE-SPECIALIZED PILOT

The exploratory pilot compares four generalists with a fixed team of two evidence explorers and two evidence verifiers on 100 uniformly sampled held-out HotpotQA questions, with three matched repetitions. Explorers seek new bridge entities and sources; verifiers check candidate claims. Both roles retain the primary Qwen2.5-14B retrieval tools, four search rounds, voting rule, and frozen routing parameters. Only role instructions change. Table 12 reports gains of 3.5 and 2.7 points on this subset, for an observed gain change of -0.8 points. This pilot uses a small fixed-composition subset.

Table 12: Role-specialization pilot on HotpotQA/Qwen2.5-14B (100 held-out questions; four agents; three matched decoding repetitions). Δ is MERGE minus SEPARABLE, in F1 points. The final row reports the change in gain between conditions on the same questions.

Team	Role composition	SEPARABLE F1	MERGE F1	Δ (pp)
Homogeneous	4 generalists	0.681	0.716	+3.5
Role-specialized	2 explorers + 2 verifiers	0.696	0.723	+2.7
Gain change	$\Delta_{\text{role}} - \Delta_{\text{hom}}$	—	—	-0.8

B.8 CONTINUATION-POLICY SENSITIVITY

We test whether the four-agent intervention profile is sensitive to the continuation policy. The comparison uses 50 held-out HotpotQA questions. For each question, we select the same logged checkpoint, selected cluster, selected agent, and matched routing context for both continuation conditions. We then run the intervention under either π_{cont} or SEPARABLE, using the same matched decoding repetitions. Thus only the policy that generates the replay schedule changes; g_ϕ is not refitted. The paired schedule, block on later delivery of the selected cluster to the selected agent, and eligible support follow the primary intervention analysis.

Table 13: Continuation-policy sensitivity at $N = 4$ on the same 50 held-out HotpotQA questions. Means and contrasts are in F1 points. C_- compares $\delta = 1$ with $\delta = 2$, and C_+ compares $\delta = 1$ with $\delta = 0$ on matched question support.

Policy	Stage	$\mu(2)$	$\mu(1)$	$\mu(0)$	C_-	C_+
π_{cont}	Early	2.60	1.50	0.96	-1.10	$+0.56$
π_{cont}	Late	0.32	1.82	0.39	$+1.51$	$+1.43$
SEPARABLE	Early	2.45	1.66	1.12	-0.81	$+0.55$
SEPARABLE	Late	0.48	1.63	0.57	$+1.16$	$+1.07$

Both continuation conditions show the same qualitative late-stage profile: the marginal gain is largest at $\delta = 1$. On this sensitivity set, the four-agent peak is therefore not specific to the reach or stage-gating terms in π_{cont}.

B.9 GROUP EFFECTS AND JOINT ROUTING

The factorial comparison uses the primary 400-question evaluation with three matched repetitions. The group effect is either retained or removed; search uses either the joint beam/compound solver or the existing greedy/single-move ablation; all other components are fixed. Table 14 completes the comparison with local-only greedy routing at 0.675 F1; the other means reuse the main study.

Writing $Y_{e,s}$ for mean F1 with group effect $e \in \{0, 1\}$ and search $s \in \{J, G\}$, the interaction in F1 points is

$$I = 100\big[(Y_{1,J} - Y_{0,J}) - (Y_{1,G} - Y_{0,G})\big].$$

Question-paired bootstrap samples retain all four outcomes together. Adding the group effect gains 3.6 points with joint routing and 1.1 with greedy routing. Conversely, joint routing gains 3.0 points with the group effect and 0.5 without it. Their interaction is $+2.5$ points with paired interval $[+1.3, +4.2]$. The positive interaction shows that joint routing increases the benefit of the learned group effect.

Table 14: Group effect $\times$ joint routing comparison, primary setting. The interaction is a question-paired difference in differences, in F1 points. Dashes denote inapplicable entries.

Objective	Joint routing F1	Greedy F1	Interaction (pp)	Paired 95% CI
With group effect	0.716	0.686	—	—
Without group effect	0.680	0.675	—	—
Interaction I	—	—	+2.5	$[+1.3, +4.2]$

B.10 MATCHING PREDICTED BENEFITS TO THE STATE

This control tests whether predicted sharing benefits need to match the current state, rather than simply provide a range of scores. At $N = 4$, we compute six relative predictions: the three search stages at $h = 2, 3$, using each cluster's current relevance. We randomly permute these six values while keeping the one-holder reference fixed at zero. The values and their frequencies within this six-cell set stay the same, but their assignments to search stage and holder count change. States can still occur at different frequencies during search.

We use permutation seeds 0, 1, and 2, fixing each mapping across all clusters and questions without retuning. Evaluation uses the primary 400 questions and three matched repetitions. We average permutation seeds within each question before the paired bootstrap comparison with MERGE.

Shuffling lowers F1 from 0.716 to 0.692 despite preserving the set of relative predicted benefits. The 2.4-point paired loss supports the importance of assigning predictions to the appropriate states, beyond providing joint routing with a range of scores. The shuffled mean is 0.3 points above the constant-effect control; these two controls were not evaluated with a paired contrast.

Table 15: Group-predictor state-shuffle control in the primary setting. Δ and paired 95% CIs are MERGE minus the row, in F1 points. The group-predictor and constant-effect rows reuse reported results.

Prediction model	F1	Δ	Paired 95% CI
Group predictor, state-aligned	0.716	—	—
Constant effect	0.689	+2.7	$[+1.4, +3.7]$
Shuffled cells (mean over permutations)	0.692	+2.4	$[+1.1, +3.6]$

B.11 EXACT MATCH ROBUSTNESS EVALUATION

Exact Match (EM) complements token-level F1 with an answer-level criterion: an answer receives one if it matches an accepted gold answer under the dataset's answer-normalization rules, and zero otherwise. Failed final answers receive zero. The supplementary evaluation uses the same final answers, questions, and three matched decoding repetitions as the F1 grid, without refitting or changing the routing configuration. Table 16 mirrors its method groups and column order. MERGE has the highest mean EM in all nine settings, with no ties. In the primary setting, it reaches 0.610 versus 0.590 for FIXED-SHAPE, a 2.0-percentage-point gain. Both FULL-COMM and RANDOM also exceed NO-COMM in mean EM across all nine settings, matching their F1 ordering.

For the primary HotpotQA/Qwen2.5-14B comparison, we average each method's EM over repetitions within a question, subtract FIXED-SHAPE from MERGE, and average the differences across questions. Multiplying by 100 gives percentage points. EM serves as a supplementary robustness comparison; F1 remains the primary answer-quality metric.

Table 16: Exact Match on the same 400 questions per setting as Table 1, on the $[0, 1]$ scale. H: HotpotQA; M: MuSiQue; W: 2WikiMHQA.

	Qwen2.5-14B			**Qwen3-14B**			**Gemma-3-12B**		
Method	**H**	**M**	**W**	**H**	**M**	**W**	**H**	**M**	**W**
Reference points									
FULL-COMM	0.557	0.315	0.558	0.604	0.352	0.601	0.541	0.307	0.535
NO-COMM	0.525	0.286	0.528	0.574	0.327	0.571	0.506	0.281	0.512
RANDOM	0.546	0.291	0.545	0.590	0.341	0.578	0.525	0.285	0.526
SINGLE-AGENT	0.507	0.279	0.515	0.552	0.316	0.552	0.498	0.270	0.505
Published routing (adapted)									
RCR-Router	0.561	0.318	0.560	0.609	0.355	0.605	0.547	0.311	0.542
AgentPrune	0.571	0.318	0.580	0.619	0.366	0.600	0.536	0.315	0.547
DYTOPO-STYLE	0.578	0.323	0.582	0.620	0.363	0.603	0.541	0.320	0.555
Controlled comparators									
SEPARABLE	0.573	0.321	0.570	0.620	0.362	0.608	0.551	0.321	0.558
FIXED-SHAPE	0.590	0.326	0.583	0.631	0.360	0.611	0.556	0.320	0.560
Ours									
MERGE	0.610	0.335	0.593	0.649	0.379	0.621	0.571	0.325	0.565

B.12 AGGREGATION-FREE OUTCOME SENSITIVITY

The primary paired-intervention analysis evaluates each matched continuation using the F1 of the final aggregated team answer. We additionally recompute the same interventions using the mean final F1 across individual agents:

$$Y_T^{\text{ind}} = \frac{1}{N} \sum_{i=1}^{N} \text{F1}(a_i, a^\star),$$

where a_i is agent i's final answer and $a^\star$ is the gold answer. For each paired intervention, we recompute the target as the difference in Y_T^{ind} between the same two matched continuations.

This analysis reuses the same logged states, selected routing decisions, replay schedules, and matched decoding repetitions as the primary intervention analysis. No additional rollout or model fitting is performed; only the terminal outcome used to score each continuation changes. Cell means and neighboring contrasts use the same eligible support as the primary analysis, and confidence intervals use the same question-paired bootstrap procedure.

The early C_{-} contrast remains negative at all three team sizes under mean individual-agent F1. The sharp four-agent late-stage peak, however, becomes much weaker. Thus, the early sharing pattern persists without final vote aggregation, while the late-stage holder-count pattern is more sensitive to the downstream decision rule.

B.13 TRANSFER TO FACT VERIFICATION

To test whether MERGE transfers beyond question answering, we evaluate the frozen HotpotQA/Qwen2.5-14B router on HoVer, a multi-hop fact-verification benchmark (Jiang et al., 2020). No HoVer examples are used to fit the local scorer, group predictor, or routing weights.

We retain the primary four-agent protocol: each agent runs four BM25 search rounds and retrieves five unseen documents per query. Each agent predicts SUPPORTED or NOT SUPPORTED; the final team label follows the same strict-majority rule and fixed tie-breakers as in the QA experiments. We evaluate 400 claims with three matched decoding repetitions and report verdict accuracy and Evidence F1 (Table 18). All controlled comparators use the same retrieval, aggregation, and communication setup.

MERGE reaches 0.744 accuracy and 0.641 Evidence F1, improving over the strongest controlled comparator, FIXED-SHAPE, by 1.3 and 1.9 points. The benefit therefore transfers from answer generation to binary fact verification, although both tasks remain Wikipedia-based retrieval settings.

Table 17: Estimated effects of sharing with one additional agent under mean individual-agent F1. C_- and C_+ are computed on the same matched support as Table 20; intervals are paired 95% CIs. All values are in F1 points.

N	**Stage**	$\mu(2)$	$\mu(1)$	$\mu(0)$	C_- **[95% CI]**	C_+ **[95% CI]**
4	Early	2.21	1.42	1.03	-0.79 $[-1.35, -0.23]$	$+0.39$ $[-0.25, +1.00]$
4	Middle	1.89	1.54	1.46	-0.35 $[-0.91, +0.20]$	$+0.08$ $[-0.46, +0.61]$
4	Late	0.78	1.12	0.91	$+0.34$ $[-0.21, +0.86]$	$+0.21$ $[-0.35, +0.73]$
6	Early	2.17	1.48	1.13	-0.69 $[-1.22, -0.17]$	$+0.35$ $[-0.20, +0.89]$
6	Middle	1.55	1.61	1.48	$+0.06$ $[-0.49, +0.60]$	$+0.13$ $[-0.39, +0.65]$
6	Late	1.03	1.24	1.01	$+0.21$ $[-0.34, +0.74]$	$+0.23$ $[-0.30, +0.76]$
8	Early	1.94	1.17	1.15	-0.77 $[-1.34, -0.21]$	$+0.02$ $[-0.52, +0.55]$
8	Middle	1.42	1.34	1.53	-0.08 $[-0.64, +0.47]$	-0.19 $[-0.74, +0.35]$
8	Late	1.00	1.12	1.29	$+0.12$ $[-0.44, +0.67]$	-0.17 $[-0.72, +0.38]$

Table 18: Zero-shot task transfer to HoVer fact verification using the frozen HotpotQA/Qwen2.5-14B router. Results use 400 claims and three matched decoding repetitions.

Method	Accuracy	Evidence F1
FULL-COMM	0.712	0.598
SEPARABLE	0.724	0.614
FIXED-SHAPE	0.731	0.622
MERGE	**0.744**	**0.641**

C UNDERSTANDING GROUP EFFECTS AND ROUTING DECISIONS

C.1 PREDICTING THE BENEFIT OF ONE ADDITIONAL AGENT

Prediction compares raw g_ϕ with the paired final-F1 difference, before subtracting the one-holder reference or summing predictions. The constant predictor uses the fitting-set mean. Each team size has 200 questions and 1,800 paired interventions; errors are averaged within questions and then equally across questions.

Table 19: Held-out prediction of the benefit of one additional agent. Errors are in F1 points; Spearman ρ measures rank association and R^2 measures prediction fit relative to target variation.

N	Questions / pairs	Model MAE	Constant MAE	Cal. slope	Spearman ρ	R^2
4	200 / 1,800	3.48	4.18	0.84	0.43	0.28
6	200 / 1,800	3.76	4.20	0.81	0.36	0.21
8	200 / 1,800	3.96	4.22	0.72	0.30	0.12

MAE improves by 0.70, 0.44, and 0.26 points at $N = 4, 6, 8$; calibration slopes fall from 0.84 to 0.81 and 0.72. Spearman correlations are positive at all three sizes, decreasing from 0.43 to 0.36 and 0.30. The corresponding R^2 values of 0.28, 0.21, and 0.12 show progressively less improvement in squared-error fit over the target-mean reference. Generalization weakens both ordering and scale agreement.

The model is fitted and selected only on four-agent development data; the 1,800 pairs at each larger team size are reserved for evaluation. These diagnostics therefore measure zero-shot extrapolation, not a multi-size supervised fit (Appendix A.5.2). Calibration and ranking require separate diagnostics: a positive rescaling preserves prediction rankings but changes the group effect's balance with local utility and cost.

Prediction fit versus decision value. R^2 evaluates raw terminal-F1 predictions across all sampled interventions. It is not a direct measure of the comparisons made by the router, and we do not evaluate within-state routing-action ranking accuracy. Reference subtraction and summation transform the raw

predictions before they enter the routing score, so prediction fit and policy value should be reported separately. The shuffled control shows that assigning predicted benefits to the appropriate states matters. Which prediction statistic mediates the end-to-end improvement remains unresolved.

The constant-increment comparison in Table 11 additionally tests whether a constant preference for giving evidence to more agents accounts for the routing gains. That control exceeds SEPARABLE by 0.9, 1.3, and 1.7 points at $N = 4, 6, 8$, whereas MERGE exceeds it by a further 2.7, 2.2, and 1.9 points. The four-agent paired interval is $[+1.4, +3.7]$. This comparison favors state-dependent sharing benefits over the constant preference. Shuffling the assignment of predictions to states preserves the six-cell set of values but loses 2.4 points relative to MERGE, with paired interval $[+1.1, +3.6]$ (Appendix B.10). This control provides direct evidence that state alignment matters for routing decisions.

C.2 MEASURED AND PREDICTED SHARING PATTERNS

Both matched continuations use the same replay schedule generated by π_{cont}. Their final F1 difference measures the effect of the selected routing decision under that fixed communication plan. The schedule-generation policy still matters: its stage gate and reach rule determine which plan is replayed.

Figure 3 compares measured outcomes with raw g_ϕ predictions on the same held-out samples, with 200 questions per stage/team-size setting. A cluster contributes only holder counts reachable from its logged pre-routing state; the $h = 1$ model reference does not restrict this support.

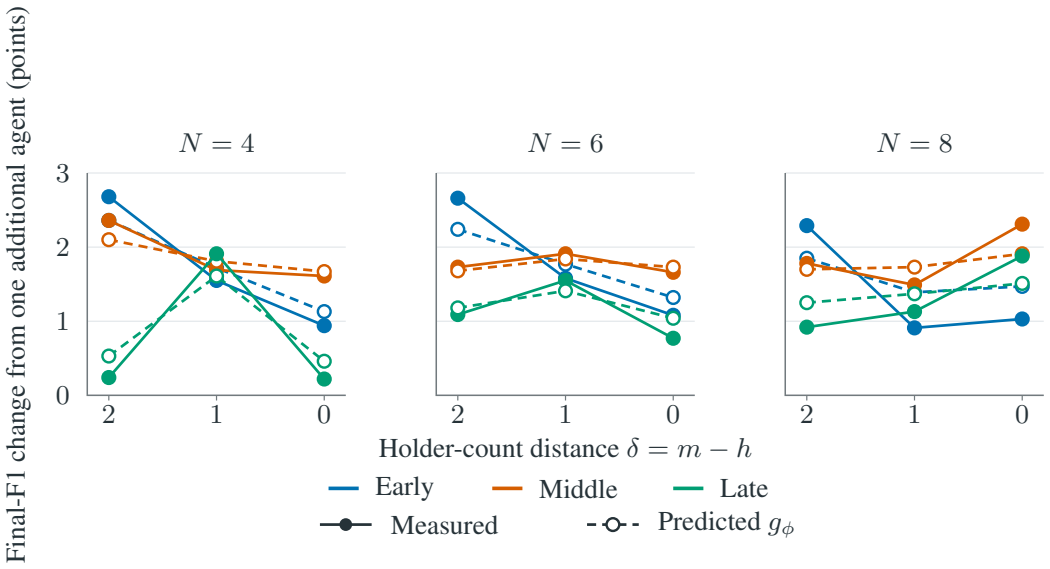


Figure 3: Estimated final-F1 changes from sharing with one additional agent across team sizes under fixed replay schedules generated by π_{cont} (F1 points). Colors denote early, middle, and late search; solid lines with filled markers show question-balanced measured effects at $\delta = 2, 1, 0$. Dashed lines with hollow markers show raw g_ϕ predictions averaged over the same held-out samples with the same question weights. Lines connect the means for each holder count. Here $\delta = m - h$ describes the holder-count feature used by the model.

The most consistent cross-size pattern is the early-stage decline from $\delta = 2$ to $\delta = 1$. The matched contrasts at $N = 4, 6, 8$ are -1.17, -1.09, and -1.39 points, and all three paired intervals lie below zero. Thus, early in search, the marginal value of sharing the same evidence with one more agent decreases as that evidence is shared more broadly.

Late-stage patterns are less uniform. At $N = 4$, the mean rises from 0.24 to 1.91 points at $\delta = 1$ and falls to 0.22 at $\delta = 0$. The matched contrasts are $+1.68$ and $+1.69$, with intervals $[0.72, 2.61]$ and $[0.79, 2.58]$. At $N = 6$, the late-stage means are 1.09, 1.55, and 0.77, again with the largest

mean at $\delta = 1$. At $N = 8$, they are 0.92, 1.13, and 1.88, with the largest mean at $\delta = 0$. Its neighboring contrasts are $C_- = +0.21$ with interval $[-0.76, +1.14]$ and $C_+ = -0.75$ with interval $[-1.71, +0.28]$; the $N = 6$ neighboring intervals also include zero. The maximizing holder count varies across settings, supporting state-dependent sharing benefits.

On the same 50-question sensitivity set at $N = 4$, both schedule-generation policies retain the late-stage maximum at $\delta = 1$ (Table 13). This suggests that the observed shape is not specific to the particular stage and reach rules used by π_{cont}.

The learned profiles reproduce the late-stage maximum at $N = 4, 6$ and the increasing trend at $N = 8$, but compress their variation. At $N = 4$, the predicted late-stage means are 0.53, 1.61, and 0.46; at $N = 8$, they are 1.25, 1.37, and 1.51. At $N = 8$, the model captures the early-stage minimum at $\delta = 1$ but misses the middle-stage dip, predicting an increase throughout instead. These cell-level comparisons show both the learned shape and the errors summarized by the ranking and calibration diagnostics. Cell means and paired contrasts use their respective eligible samples and are rounded separately. The table and figure report the same cell means.

Table 20: Estimated final-F1 changes from sharing with one additional agent under π_{cont} replay schedules (F1 points), grouped by $\delta = m - h$. C_- and C_+ compare $\delta = 1$ with $\delta = 2$ and $\delta = 0$ on matched support; n_- and n_+ count matched pairs. Intervals are paired 95% CIs.

N	Stage	$\mu(2)$	$\mu(1)$	$\mu(0)$	C_- [95% CI]	C_+ [95% CI]	n_-	n_+
4	Early	2.68	1.55	0.94	$-1.17\ [-2.02, -0.31]$	$+0.62\ [-0.24, 1.48]$	188	181
4	Middle	2.36	1.69	1.61	$-0.68\ [-1.52, 0.16]$	$+0.11\ [-0.62, 0.82]$	190	184
4	Late	0.24	1.91	0.22	$+1.68\ [0.72, 2.61]$	$+1.69\ [0.79, 2.58]$	192	187
6	Early	2.66	1.58	1.08	$-1.09\ [-1.91, -0.31]$	$+0.56\ [-0.29, 1.46]$	183	176
6	Middle	1.73	1.91	1.66	$+0.19\ [-0.69, 1.08]$	$+0.24\ [-0.58, 1.12]$	185	179
6	Late	1.09	1.55	0.77	$+0.52\ [-0.26, +1.28]$	$+0.74\ [-0.17, +1.59]$	189	181
8	Early	2.29	0.91	1.03	$-1.39\ [-2.28, -0.57]$	$-0.12\ [-1.01, 0.88]$	177	170
8	Middle	1.78	1.49	2.31	$-0.31\ [-1.12, 0.49]$	$-0.79\ [-1.61, 0.08]$	180	173
8	Late	0.92	1.13	1.88	$+0.21\ [-0.76, +1.14]$	$-0.75\ [-1.71, +0.28]$	184	177

C.3 Trace-Level Case Studies

The aggregate analyses above show that state-dependent group effects and joint routing both affect routing quality, but they do not make individual routing decisions easy to inspect. We therefore present three qualitative cases illustrating different parts of the MERGE mechanism. Case A shows how repeated sharing of the same evidence can remain useful while losing priority to complementary evidence. Case B shows how two routing decisions that are unattractive individually can form a useful joint routing action. Case C shows how a selected routing decision can change subsequent search before affecting final team performance.

These examples illustrate the mechanisms rather than estimate their population frequency. Evidence descriptions are paraphrased for readability where noted; the case-study figures use the same evidence clusters, holder counts, communication costs, and paired interventions as the main method.

Routing decisions. Figure 4 illustrates two ways in which group-aware routing changes the selected routing action.

Case A: repeated sharing remains useful but loses priority. Early in search, evidence cluster c is available to one agent and is locally useful to Agents 2 and 3. A different cluster d supplies a complementary geographic bridge for Agent 3. Clusters c and d compete for the same routing opportunity for Agent 3. Under the local-only objective, each agent is scored independently: c ranks ahead of d for Agent 3, so both copies of c are sent and d is not selected.

After one copy of c has already been selected, MERGE evaluates the next same-cluster routing decision after another agent already holds the cluster. Its group adjustment reduces its total marginal value, although the routing decision itself can still remain useful. For Agent 3, cluster d therefore becomes preferable to another copy of c.

The point is not that repeated sharing is harmful. Rather, its marginal value can decline relative to competing evidence as the same evidence is shared more broadly. This gives a trace-level illustration of the early-stage pattern reported in Section 6.4.

Case B: neither individual routing decision changes the team answer, but the pair does. In the second example, Agent 1 has evidence cluster c, which supports the correct answer W, while the other three agents currently answer X. Without further sharing, the team therefore outputs X.

Sending c to Agent 2 alone changes that agent's answer from X to W, but the resulting 2–2 tie is still resolved to X by the fixed tie-breaking rule. Sending c only to Agent 3 produces the same outcome. Thus, either individual routing decision changes one agent's answer but not the final team answer.

Each routing decision also has a small negative objective gain when considered alone, so greedy routing selects neither. When they are evaluated together, the second copy of c receives a positive group adjustment that outweighs both small losses, so the pair has positive total gain and joint routing selects both routing decisions. Agents 2 and 3 both switch to W, producing a 3–1 vote and changing the final team answer from X to W. This case illustrates complementarity within a routing action.

Downstream effects of paired interventions. The previous two cases concern online routing decisions. Case C instead illustrates what can be captured by the paired-intervention target. The question asks what the current conference of the Vermont Catamounts men's soccer team was called from 1988 to 1996. Answering it requires two hops: the team's current conference, the America East Conference, and that conference's former name, the North Atlantic Conference.

Figure 5 compares two matched continuations that start from the same logged state and replay the same subsequent communication schedule. Agent 1 already knows the team's current conference, whereas Agent 3 does not. The only intervention-time difference is whether Agent 3 receives this selected fact. Without it, Agent 3 keeps searching from the team-level page, and its next search does not reach the conference's former name. With it, Agent 3's next query instead targets the former name of the America East Conference and retrieves the missing second-hop evidence. The first observable divergence therefore occurs in the next search query, before final answer aggregation.

Averaged over three matched decoding repetitions, mean final team F1 increases from 0.31 to 0.68, a difference of $+0.37$. This example shows that the intervention target can capture effects mediated by downstream query formulation and evidence retrieval, rather than only the direct effect of informing one additional agent before final aggregation.

Case A: repeated sharing remains useful but loses priority.

Question

Q: Whose **navigator father** explored the **east coast** of the continent where **Ignacio Esparza** was later born?
Clue ***c***: "John Cabot explored the east coast of North America for England."
Clue ***d***: "Mexico is in North America."
Chain: Ignacio Esparza → Mexico → North America → John Cabot → Sebastian Cabot

Team state (early search)

The team has already established that **Ignacio Esparza was born in Guadalajara, Mexico.**
Agent 1 has clue *c*; another agent has clue *d*.
Clue *c* is useful to both Agent 2 and Agent 3, while clue *d* provides a complementary bridge for Agent 3.
For Agent 3, ***c* and *d* compete for the same routing opportunity**: only one of them is selected for Agent 3 in this routing step.

Local-only router

For Agent 2, clue *c* has the highest local value → send.
For Agent 3, clue *c* also ranks above clue *d* locally → send *c*.
Choosing *c* for Agent 3 leaves the complementary clue *d* unselected.

MERGE

Once *c* → Agent 2 is selected, sending *c* to Agent 3 would **give the same clue to a third agent.** MERGE's group adjustment discounts this repeated exposure, though it remains useful.
Selecting *c* → Agent 2 flips Agent 3's ranking: *d* now outranks *c*, so Agent 3 receives *d* instead.

Case B: neither routing decision changes the answer alone, but together they do.

Question

Q: Who was the **paternal grandfather** of Anna Dorothea, Abbess of Quedlinburg?
W William, Duke of Saxe-Weimar ✓
X John Ernest II (her father) ✗

Team state (late search)

Only Agent 1 holds clue ***c***, which **supports W.**
The other three agents answer X.

Greedy (one decision at a time)

c → Agent 2 alone: small negative gain
c → Agent 3 alone: small negative gain
Both individual decisions are negative, so greedy sends nothing.

MERGE (decisions scored jointly)

The first copy of *c* gets no group term; the second copy gets a **positive group term.**
Pair: the group term outweighs both small losses, so send *c* to Agents 2 and 3.

Final vote

	A1	A2	A3	A4	
No routing	W	X	X	X	3–1 → X ✗
c → A2	W	W	X	X	2–2 → X ✗
c → A3	W	X	W	X	2–2 → X ✗
c → A2, A3	W	W	W	X	3–1 → W ✓

Alone, each copy only makes a 2–2 tie, and the tie-breaker still picks X. Together they make a 3–1 vote for W.
All four matched continuations start from the same logged state and replay the same communication schedule. Votes are outcomes, not router inputs.

Figure 4: **Trace-level routing examples. Case A:** after clue c is sent to Agent 2, sending c to Agent 3 remains useful but is displaced by the complementary clue d once the group adjustment is applied. **Case B:** only Agent 1 holds clue c. Each individual routing decision has a small negative objective gain and, in the matched continuations, leaves a 2–2 tie that the fixed tie-breaker resolves to the wrong answer; the joint routing action has positive gain and yields a correct 3–1 vote.

Case C: the selected routing decision changes downstream search before final aggregation.

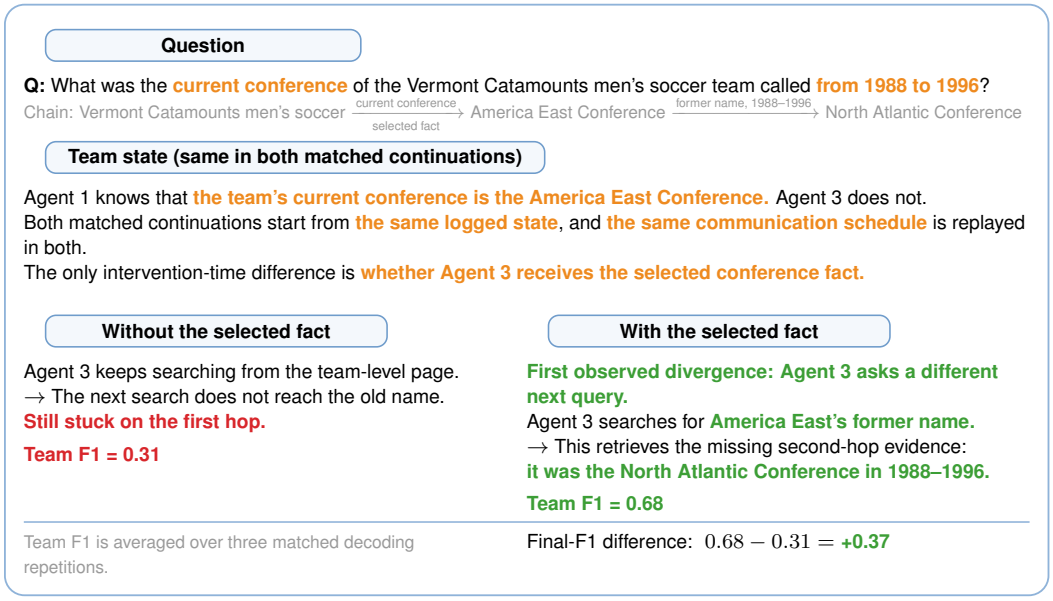


Figure 5: **A paired intervention with a downstream search-mediated effect.** Both matched continuations start from the same logged state and replay the same communication schedule; they differ only in whether Agent 3 receives the team's current conference. With this selected fact, Agent 3 changes its next query and retrieves the conference's former name. Mean final team F1 increases from 0.31 to 0.68, a $+0.37$ difference.

D STATISTICAL PROTOCOL AND BASELINE ADAPTATIONS

D.1 STATISTICAL PROTOCOL AND SCOPE

The statistical unit is a question. We average three matched decoding repetitions within each question, then jointly resample question IDs 10,000 times across methods. Failed final answers retain zero F1 and all consumed resources. Policy contrasts use MERGE minus comparator; ablations use variant minus full. Group-predictor errors give each question equal weight. Intervention resampling keeps a question's states and repetitions together; neighboring contrasts compare $\delta = 1$ against $\delta = 2$ and $\delta = 0$ using eligible matched samples. These samples can differ from those used for the separately reported cell means. Paired final-F1 differences measure outcome changes under a shared replay schedule; the routing score uses predictions relative to the one-holder reference.

The primary controlled comparison is against FIXED-SHAPE; adapted published routers provide external reference points. Because the grid selects the largest non-MERGE mean in each cell, Table 9 reports point contrasts without intervals. Intervals reported elsewhere are pointwise and not adjusted for multiplicity. The supplementary primary EM contrast is mean-only. The aggregation-free neighboring contrasts in Table 17 use the same paired interval procedure as the primary intervention analysis. Follow-up controls, non-primary grid cells, and the role pilot are exploratory.

Grid-wide direction and multiplicity. All nine displayed margins are positive. An exact sign test under independent cells gives $p = 0.004$ (two-sided; directional $p = 0.002$). We report this as a sensitivity check because cells share fitted components and datasets.

Published routers are adapted to the common search environment rather than reproduced as their original systems. Prediction and intervention diagnostics characterize the benefit of sharing with one additional agent, not the benefit of every selected routing decision. Each main-grid and additional-backbone setting uses 400 questions; additional-backbone results report mean performance. Appendix D.3 derives the offline continuation workload from the retained pair counts, separately from measured online token use and latency.

D.2 ADAPTED PUBLISHED ROUTERS

RCR-Router adapts role- and stage-aware memory selection to four search agents, common evidence items, and transmitted-evidence caps (Liu et al., 2025). AgentPrune adapts message/link pruning from its original multi-task settings to retrieval QA and final F1 (Zhang et al., 2025). DYTOPO-STYLE retains receiver-need/sender-evidence matching but removes DyTopo's manager and manager-conditioned round goals, holding orchestration and aggregation fixed (Lu et al., 2026).

Manager-conditioned goals change what agents search for and which evidence enters the candidate pool. Including a manager only for DyTopo would change both the search process and evidence routing. We instead fix task instructions, retrieval, and the final vote, and evaluate its receiver–sender matching component in the same environment as the other routers. The resulting DYTOPO-STYLE comparison concerns evidence routing, not the complete manager-guided system.

The single agent shares the four-round retrieval procedure and a 16-query ceiling, while actually making four queries. NO-COMM runs four independent search trajectories with the same vote. RANDOM uniformly permutes feasible candidate routing decisions at each step and adds them in that order subject to the common routing constraints; its seeds are fixed before evaluation. Their F1 reference points appear in Table 1. Table 5 reports the common routing configuration.

D.3 SAMPLE COUNTS AND EXCLUSIONS

After filtering for eligibility, fitting, model selection, and prediction evaluation use fixed, pre-specified quotas balanced over search stage, pre-routing holder count and target holder count, within reachable cells. States without an eligible cluster are recorded as `no_eligible_cluster`; eligibility does not require a cluster held by only one agent. Table 21 counts paired configurations before averaging decoding repetitions. Routing evaluation instead counts questions.

Replay skips occur when an agent already has the scheduled item or the scheduled routing decision is no longer admissible under the replayed state. These execution-time checks are distinct from the

Table 21: Collection counts before averaging decoding repetitions. Excluded equals considered minus kept, including quota filtering, restoration, infeasibility, missing outcomes, and matching failures.

Partition	Questions	Considered	Eligible	Kept pairs	Excluded
Discovery	300	3,120	2,970	2,700	420
Fitting	1,000	9,950	9,510	9,000	950
Selection	250	2,550	2,425	2,300	250
Prediction $N = 4$	200	2,010	1,905	1,800	210
Prediction $N = 6$	200	2,055	1,918	1,800	255
Prediction $N = 8$	200	2,120	1,930	1,800	320
Policy confirmation	400	—	—	—	—

pre-collection exclusions in Table 21; the exclusion totals are not replay-skip counts. Blocking later delivery of the selected cluster to the selected agent is a separate protocol rule.

Offline collection workload. Fitting and selection retain $9,000 + 2,300 = 11,300$ pairs. Counting two matched continuations and three matched decoding repetitions per pair gives $2 \times 3 \times 11,300 = 67,800$ checkpoint-to-answer team continuations. Including discovery and the three prediction sets gives $2 \times 3 \times (2,700 + 9,000 + 2,300 + 3 \times 1,800) = 116,400$. These protocol-based estimates count checkpoint-to-answer continuations before continuation reuse and retries, rather than full runs from the initial question. Continuation lengths vary with the intervention stage. These continuation counts do not determine token cost and do not measure the cost of generating replay schedules. Prefix collection, excluded attempts, and model fitting add cost; reused continuations reduce duplicated execution. The resulting workload is of order 10^5 matched continuations under separate execution.

Amortized offline cost. Let C_{off} denote the offline token cost of intervention collection and fitting, and let M be the number of deployment questions. The amortized token use of MERGE is

$$11{,}274 + \frac{C_{\text{off}}}{M}$$

tokens per question. Relative to FIXED-SHAPE, which uses 1,506 more online tokens per question, the online saving offsets this offline cost when

$$M \geq \frac{C_{\text{off}}}{1{,}506},$$

excluding the comparator's own offline training cost. Table 2 reports online resource use. Total cost depends on C_{off} and the deployment volume.