REDUCING COORDINATION FAILURES IN LLM AGENT POPULATIONS VIA BILEVEL INFORMATION ALLOCATION

Anonymous authors
Paper under double-blind review

ABSTRACT

When multiple LLM agents act separately on a shared task, individually reasonable decisions can produce poor collective outcomes: agents may crowd the same scarce resources, duplicate effort, or leave useful alternatives unused. We ask whether these failures can be reduced without updating the acting agents, by changing only which truthful state information each agent receives. The problem is difficult because responses to information are unknown and history-dependent, evaluating an allocation requires costly agent execution, and what is useful for one agent depends on what the others see. We introduce BRIA, a Bilevel Response-guided Information Allocation framework with two alternating levels. BRIA learns how the agent population responds to joint information allocations and uses that response model to choose better allocations. The two levels form a closed loop: the learned response guides allocation, the resulting allocations change the histories subsequently reached, and randomized data from those histories are used to update the response model. Across three datasets, BRIA improves response prediction and end-to-end coordination while keeping the amount of information per agent fixed, with consistent reductions in crowd-out. Ablations support history modeling, joint allocation, and repeated response-model updates, showing that jointly optimizing truthful information allocation can improve coordination without updating the acting agents.

1 INTRODUCTION

When several LLM agents act on a shared task, improving each agent in isolation does not guarantee a good collective outcome. Agents can make individually plausible choices and still crowd the same scarce resource, duplicate effort, or leave useful alternatives unused. Recent studies of LLM-agent populations likewise show that collective behavior depends on interaction structure, repeated context, and coordination among agents (Park et al., 2023; Ashery et al., 2025; Agashe et al., 2025; Zhu et al., 2025; Mi et al., 2025). The source of failure is therefore not only individual reasoning quality: what is sensible for one agent can become wasteful when many agents respond similarly to the information they receive.

Most work on LLM-agent coordination changes the agents themselves—through learned collaborators, communication structure, agent participation, or topology (Sun et al., 2026; Wang et al., 2025b;a; Shen et al., 2025). Retrieval and memory methods instead decide what information is useful for one recipient (Shinn et al., 2023; Ning et al., 2026). Shared-resource tasks expose a different question: *who should see what?* A fact can be individually relevant yet collectively harmful if it steers many agents toward the same scarce option (Figure 1).

We therefore study *joint information allocation*: the allocator chooses which truthful optional state facts each agent observes, while each agent receives the same fixed number of facts and the acting agents themselves are unchanged. The central difficulty is that agent responses to information are unknown *a priori*. Consequently, the allocation that produces the best collective outcome cannot be specified in advance and must instead be learned from execution. This viewpoint is related to information-design formulations that optimize what receivers observe (Kamenica & Gentzkow,

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.

2011; Bergemann & Morris, 2019), but here the response of an LLM-agent population must itself be learned.

Three difficulties make this nontrivial. First, learning which allocations work requires executing the population, so direct search is expensive; learned-model rollouts offer a way to reuse costly interaction data for planning (Lin et al., 2024a; Tang et al., 2024; Gu et al., 2025). Second, responses depend on interaction history, which changes as the system evolves (Mi et al., 2025). Third, allocations are coupled across agents: the value of showing a fact to one agent depends on what the others see. The key opportunity is that one execution reveals the actions of the whole population. A learned response model can therefore reuse that execution to evaluate many additional allocations without repeatedly calling the agents.

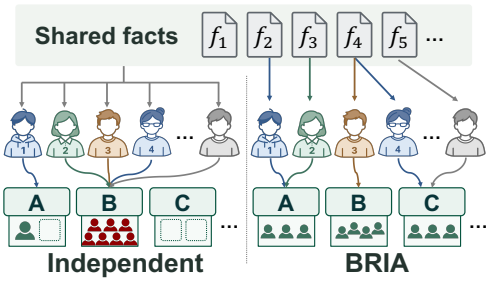


Figure 1: **Why joint allocation matters.** Independent information selection can crowd agents onto scarce options, while joint allocation coordinates what different agents observe.

This motivates BRIA (**B**ilevel **R**esponse-guided **I**nformation **A**llocation). BRIA alternates between two coupled levels: a response-learning level that models how the population reacts to joint information allocations, and an allocation level that uses the fitted model to choose allocations with low predicted task loss. The two levels form a closed loop: the allocator changes which histories are reached, and randomized observations from those histories are then used to update the response model. We use *bilevel* to describe these coupled optimization levels, which are updated alternately rather than through differentiation of an exact nested program.

Contributions. Our main contributions are:

1. We identify *joint information allocation* as a distinct coordination problem: collective behavior can be improved by optimizing who sees what, even when the acting agents themselves are unchanged.
2. We show that population response modeling enables efficient allocation optimization: one real execution can support the evaluation of many unexecuted joint allocations.
3. We derive a practical response-guided optimization principle for joint allocation, linking a KL-regularized target distribution to an autoregressive proposal update under evolving interaction histories.

2 Related Work

LLM multi-agent coordination. LLM-agent systems have been studied for social simulation, repeated interaction, collaborative reasoning, and explicit coordination (Park et al., 2023; Ashery et al., 2025; Agashe et al., 2025; Zhu et al., 2025; Riedl, 2026; Mi et al., 2025). Recent methods improve collaboration by training specialized collaborators or changing which agents communicate and participate (Sun et al., 2026; Wang et al., 2025b;a; Shen et al., 2025). These works show that interaction structure and information flow matter, but they do not directly optimize how a limited set of truthful state facts should be divided across agents that compete for shared resources. BRIA optimizes that allocation directly. MF-LLM (Mi et al., 2025) is especially related because it uses an information bottleneck and a fixed pretrained prior to compress population history for behavioral simulation. BRIA adapts this idea to an allocation-conditioned response model: the bottleneck is conditioned on the current state, and the learned response is used to compare candidate information allocations.

Context selection and information design. Retrieval, memory, and observation-selection methods choose useful context using relevance, diversity, or learned utility (Carbonell & Goldstein, 1998; Shinn et al., 2023; Ning et al., 2026; Wang et al., 2025a). They usually select context for one recip-

ient at a time, so individually useful contexts can still send several agents toward the same option. Classical information design instead optimizes what information receivers observe (Kamenica & Gentzkow, 2011; Bergemann & Morris, 2019), including settings with congestion effects (Griesbach et al., 2024). In those models the receiver response follows from the game; for LLM agents it must be learned from execution. BRIA combines these ideas by learning how the population responds to an allocation and then choosing information jointly across agents.

Model-based planning. World-model methods use learned dynamics and imagined rollouts to reduce costly interaction (Lin et al., 2024a; Tang et al., 2024; Gu et al., 2025). BRIA uses the same broad idea for a different prediction target. Instead of mainly predicting how the environment changes after an action, it predicts how the agents change their actions when they receive different information. This lets us compare many joint allocations before executing one with the real agents.

3 Problem Formulation

At round t, the allocator observes the current public state X_t and the public history $\mathcal{C}_t = (X_{<t}, K_{<t}, A_{<t}, \ell_{<t})$. Each acting agent i may also have private memory H_i^t, which is maintained by the agent system and is not available to the allocator.

For agent i, let $\mathcal{F}_i(X_t)$ be the truthful optional facts that can be shown at the current round, with $|\mathcal{F}_i(X_t)| \geq \kappa$. The allocator chooses exactly κ facts,

$$K_i^t \subseteq \mathcal{F}_i(X_t), \qquad |K_i^t| = \kappa, \tag{1}$$

and the joint allocation is $K_{1:n}^t = (K_1^t, \ldots, K_n^t)$. We write $\mathcal{K}_t(X_t)$ for the set of all valid joint allocations.

After receiving its allocated facts, agent i acts according to

$$A_i^t \sim \pi_i\big(\cdot \mid H_i^t, r_{\text{obs}}(X_t, K_i^t)\big)\,, \tag{2}$$

where r_{obs} renders the state and selected facts into the agent context. Across allocation methods, the acting agents, prompts, decoding rules, memory updates, feasible actions, and environment rules are unchanged. Only $K_{1:n}^t$ changes.

Let the round-level task loss be $\ell_t = \ell(X_t, A_{1:n}^t)$, where ℓ is determined by the task rules and the realized joint action. Our goal is to choose an allocation policy Γ that maps the public interaction state to a distribution over valid joint allocations and minimizes cumulative task loss,

$$\min_{\Gamma}\ \mathbb{E}_{\Gamma}\left[\sum_{t=1}^{T} \ell_t\right], \qquad K_{1:n}^t \in \mathcal{K}_t(X_t). \tag{3}$$

Direct search is expensive because evaluating one allocation requires executing the whole agent population. BRIA therefore learns how agents respond to information and uses that model to compare allocations before executing one.

4 Bilevel Response-guided Information Allocation

Figure 2 summarizes BRIA as a closed-loop two-level procedure. The response-learning level fits a history-aware population response model, and the allocation level uses that model to improve a joint allocation proposal through model-based evaluation. The allocation update changes the histories visited by the population; new randomized observations on those histories then update the response model. Thus neither level is trained once and frozen: the two levels are updated alternately and coupled through the interaction data they induce.

Direct optimization uses only the realized loss of an executed allocation, whereas the response model reuses each population execution to evaluate many candidate allocations before execution.

4.1 History-aware population response model

Latent history representation. Let $\mathcal{C}_t$ denote the public interaction history and X_t the current public state. We introduce a latent summary

$$M_t \sim q_\phi(\cdot \mid \mathcal{C}_t, X_t), \tag{4}$$

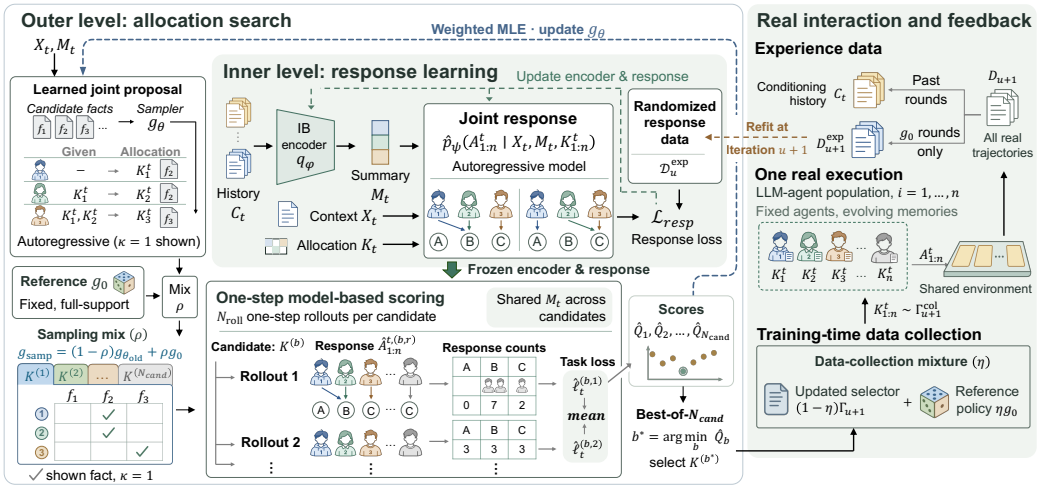


Figure 2: **Closed-loop bilevel alternating updates in BRIA.** The response-learning level learns how the population responds to information allocations from randomized data. The allocation level proposes joint allocations, scores them with model-based rollouts, and updates the proposal toward lower predicted task loss. The best-scoring allocation is executed, which changes the subsequent history; new randomized g_0 rounds then refresh the response model. BRIA does not update the acting agents.

which represents the part of the history that remains useful for predicting current responses. The summary is an auxiliary variable used by BRIA and is not part of the acting agents' observations.

Definition 1 (allocation-conditioned population response). For parameters $\lambda = (\phi, \psi)$, BRIA represents the response of the population to a candidate allocation by

$$\widehat{P}_\lambda(A^t_{1:n} \mid \mathcal{C}_t, X_t, K^t_{1:n}) = \mathbb{E}_{M_t \sim q_\phi(\cdot \mid \mathcal{C}_t, X_t)} \left[\widehat{p}_\psi(A^t_{1:n} \mid X_t, M_t, K^t_{1:n}) \right]. \tag{5}$$

The outer expectation marginalizes the compressed history representation; the conditional model inside it describes how the population reacts once that summary and the candidate allocation are specified. We parameterize this conditional distribution by the chain rule,

$$\widehat{p}_\psi(A^t_{1:n} \mid X_t, M_t, K^t_{1:n}) = \prod_{i=1}^{n} \widehat{p}_{\psi,i}\left(A^t_i \mid X_t, M_t, K^t_{1:n}, A^t_{<i}\right). \tag{6}$$

Remark 1. Equation equation 6 is a factorization of a centralized predictor, not a communication protocol among the acting agents. By the probability chain rule, any joint response distribution can be written in this form under a chosen ordering.

Conditional information bottleneck. The history summary should preserve information that predicts the population response while discarding irrelevant history. Following the information bottleneck principle (Alemi et al., 2017) and the fixed-prior construction used for population summaries in MF-LLM (Mi et al., 2025), we start from

$$\mathcal{J}_{\text{IB}} = \beta_{\text{IB}}\, I(M; \mathcal{C} \mid X) - I(M; A_{1:n} \mid X, K_{1:n}). \tag{7}$$

The first term compresses history beyond what is already known from the current state X; the second retains historical information that helps predict the population response once the allocation $K_{1:n}$ is given.

Applying the standard variational bounds with a fixed prior $r_0(M \mid X)$ gives the trainable response loss

$$\begin{aligned} \mathcal{L}_{\text{resp}}(\phi, \psi) = & -\mathbb{E}_{\mathcal{D}^{\text{exp}},\, M \sim q_\phi} \log \widehat{p}_\psi(A_{1:n} \mid X, M, K_{1:n}) \\ & + \beta_{\text{IB}}\, \mathbb{E}_{\mathcal{D}^{\text{exp}}} D_{\text{KL}}\left(q_\phi(M \mid \mathcal{C}, X) \,\|\, r_0(M \mid X)\right). \end{aligned} \tag{8}$$

The first term fits the joint response model; the second is the variational information-bottleneck penalty on information carried from history beyond the current state. We keep r_0 fixed. Appendix A gives the derivation.

Randomized response data. The response model must learn how agents react to *different* information allocations. If it were fitted only on allocations chosen by the current selector, the training data would increasingly concentrate on allocations that the selector already prefers. BRIA therefore reserves a subset of rounds for randomized allocation. A reference policy g_0 assigns nonzero probability to every valid allocation,

$$g_0(k \mid X) > 0, \qquad \forall k \in \mathcal{K}_t(X). \tag{9}$$

Only these randomized g_0 rounds are used to fit the response model. Other executed rounds still affect the subsequent interaction history. This keeps the response model exposed to diverse allocations as the allocator evolves.

4.2 Response-guided joint allocation

The outer level uses the learned response model to improve information allocation. Given the current state and history summary, it repeatedly performs the same operation: propose several joint allocations, predict how the population would respond to each one, estimate the resulting task loss, and update the proposal toward allocations with lower predicted loss.

Proposing joint allocations. Let g_θ be the allocation proposal. Rather than choosing information for each agent independently, it constructs the population allocation autoregressively:

$$g_\theta(K_{1:n} \mid X, M) = \prod_{i=1}^{n} g_{\theta,i}(K_i \mid X, M, K_{<i}). \tag{10}$$

Thus the information assigned to a later agent can depend on what has already been assigned to earlier agents. Each factor outputs a valid κ-fact subset.

Example 1 (why joint proposals matter). Consider $n \geq 2$ agents and n unit-capacity options, where every agent receives one fact highlighting one option and every agent has a uniform marginal over options. Independent assignments cover

$$\mathbb{E}[N_{\text{cov}}] = n\left[1 - \left(1 - \frac{1}{n}\right)^n\right] < n \tag{11}$$

distinct options in expectation. A uniformly random permutation has the same marginal for every agent but covers all n options with probability one. Thus joint dependence can improve population coverage even when each agent has the same marginal allocation. Appendix E gives the calculation.

Scoring candidate allocations. For each candidate joint allocation k, BRIA estimates the task loss that would result if the population received that allocation. In a model rollout, $\widehat{\ell}_{t+h} = \ell(\widehat{X}_{t+h}, \widehat{A}_{1:n}^{t+h})$ applies the same task-loss function as the real environment to the model-predicted joint action (and predicted state when $h > 0$). We use

$$\overline{Q}_t(k \mid m) = \mathbb{E}\left[\sum_{h=0}^{T_{\text{p}}-1} \widehat{\ell}_{t+h} \;\middle|\; X_t, M_t = m, K_t = k\right] \tag{12}$$

as its predicted task loss. Responses inside the rollout are sampled from the learned joint response model in Eq. equation 6. The same history summary m is used for all candidates at the decision point. When $T_{\text{p}} = 1$, the score reduces to the predicted immediate task loss. For $T_{\text{p}} > 1$, the expectation is taken under a specified continuation policy together with a state/history transition model. All candidates are evaluated under the same continuation rule.

With N_{roll} rollouts for candidate $K^{(b)}$, BRIA estimates

$$\widehat{Q}_b = \frac{1}{N_{\text{roll}}} \sum_{r=1}^{N_{\text{roll}}} \sum_{h=0}^{T_{\text{p}}-1} \widehat{\ell}_{t+h}^{(b,r)}. \tag{13}$$

At execution time, the candidate with the smallest estimated loss is selected:

$$\widehat{K}_t = K^{(\arg\min_b \widehat{Q}_b)}. \tag{14}$$

Only $\widehat{K}_t$ is sent to the acting agents. Because the executed allocation affects the next interaction history, this selection step also changes which histories the response model will encounter later.

Improving the proposal from predicted outcomes. The rollout scores provide a direct learning signal for the allocator: joint allocations predicted to produce lower task loss should become easier to sample in future rounds. For a fixed context, write $Q(k) = \overline{Q}_t(k \mid m)$. We define an ideal target distribution that trades off predicted task loss against deviation from the reference policy g_0:

$$g^* \in \arg\min_g \{\mathbb{E}_{K\sim g}[Q(K)] + \tau D_{\mathrm{KL}}(g \,\|\, g_0)\}. \tag{15}$$

Proposition 1 (optimal KL-regularized proposal). For a fixed context, a finite valid allocation set, $\tau > 0$, and full-support g_0, Eq. equation 15 has the unique solution

$$g^*(k) = \frac{g_0(k)e^{-Q(k)/\tau}}{Z}, \qquad Z = \sum_{k'\in\mathcal{K}_t(X)} g_0(k')e^{-Q(k')/\tau}. \tag{16}$$

Thus the target keeps support from g_0 while exponentially favoring allocations with lower predicted loss. Appendix B gives the derivation.

Computing g^* exactly is infeasible because the joint allocation space is combinatorial. In practice, BRIA uses g^* as the motivation for a finite-candidate proposal update: it samples candidate allocations from a mixture of the current proposal and the reference policy g_0, evaluates each candidate with the predicted task loss $\widehat{Q}_b$, and assigns larger weights to candidates with lower predicted loss. The mixture retains a randomized reference component in candidate sampling, while an importance correction accounts for the candidate-sampling distribution; Appendix B gives the finite-candidate weights and shows that this correction is bounded.

The proposal is then fitted to the weighted candidate set:

$$\mathcal{L}_{\mathrm{alloc}}(\theta;\lambda) = -\mathbb{E}\sum_{b=1}^{N_{\mathrm{cand}}} \mathrm{stopgrad}(w_b)\log g_\theta(K^{(b)} \mid X, M). \tag{17}$$

Here w_b is the normalized candidate weight defined in Appendix B. The dependence on λ enters through the response-model scores used to construct these weights. In practice, the update is implemented by sampling candidates, scoring them with the response model, and increasing the probability of candidates with better predicted outcomes.

4.3 Closed-loop bilevel alternating updates

The two levels form a feedback loop. The response-learning level learns how the population responds to information, and the allocation level uses that model to improve the allocation proposal. Although only randomized g_0 rounds enter $\mathcal{D}_u^{\mathrm{exp}}$, the histories in those rounds are generated by the current allocation process. Therefore, changing the allocator changes the history distribution on which the next response model is fitted.

Let $\lambda = (\phi, \psi)$. At outer iteration u,

$$\lambda_u^* \in \arg\min_\lambda \mathcal{L}_{\mathrm{resp}}(\lambda; \mathcal{D}_u^{\mathrm{exp}}), \tag{18}$$

$$\theta_{u+1} \in \arg\min_\theta \mathcal{L}_{\mathrm{alloc}}(\theta; \lambda_u^*). \tag{19}$$

The first update fits the response model from randomized response data; the second updates the proposal using candidate scores produced by that fitted model. These are alternating updates between two coupled levels, not differentiation through an exact nested bilevel optimum.

To keep collecting response data as the allocator evolves, BRIA uses the reference policy g_0 on a fraction η of collection rounds and the updated selector on the remaining rounds. Only the randomized g_0 rounds are added to $\mathcal{D}_{u+1}^{\mathrm{exp}}$ for the next inner update. All executed rounds still affect the subsequent history $\mathcal{C}_{t+1}$. Algorithm 1 therefore repeats a simple cycle: fit the response model, improve the allocation proposal, collect new interaction data, and refit.

5 Experiments

We evaluate two questions that follow directly from the method: **RQ1**, can the learned response model predict how the population reacts to information allocations; and **RQ2**, does optimizing

Algorithm 1 BRIA training

collect initial randomized data $\mathcal{D}_0^{\text{exp}}$ with g_0
initialize q_ϕ, $\widehat{p}_\psi$, and g_θ; keep r_0 fixed
for $u = 0, 1, \ldots, U-1$ **do**
  fit $\lambda_u^* = (\phi, \psi)$ using $\mathcal{D}_u^{\text{exp}}$ by Eq. equation 18
  sample contexts and N_{cand} joint allocations from the proposal–reference mixture
  estimate $\widehat{Q}_b$ by N_{roll} model-based rollouts of horizon T_{p}
  compute the normalized candidate weights w_b and update g_θ by Eq. equation 17
  collect new rounds, using g_0 on an η fraction and the updated selector otherwise
  form $\mathcal{D}_{u+1}^{\text{exp}}$ by adding only new g_0-branch transitions
end for
return the final best-of-N selector

those allocations improve outcomes when the real agents are executed? We then ablate the main components.

5.1 Setup

Tasks. We conduct experiments on three real-world public datasets. **Citi Bike** contains trip and station-level records from a large-scale urban bike-sharing system (Citi Bike, 2026). **LaDe** contains real-world last-mile delivery records collected from an industrial logistics platform (Wu et al., 2024). **AskUbuntu** contains question and interaction records from the Ask Ubuntu online community (Lei et al., 2016). The acting agents use Qwen3-8B-AWQ (Yang et al., 2025; Lin et al., 2024b); the auxiliary response model uses Qwen2.5-0.5B-Instruct (Yang et al., 2024). All allocation methods use the same acting agents and task conditions. Chronological splits and training details are given in Appendix G and Appendix H.

Baselines. RQ2 includes information controls, non-learned allocation methods, learned allocation methods, and stronger-agent controls. Standard comparisons include MMR (Carbonell & Goldstein, 1998), DPP (Kulesza & Taskar, 2012), Joint REINFORCE (Williams, 1992), MAPPO (Yu et al., 2022), and QMIX (Rashid et al., 2018), together with CoT (Wei et al., 2022) and ReAct (Yao et al., 2023) controls. All allocation methods in this comparison give each agent the same number of optional facts. Appendix I defines every comparison.

Metrics. For RQ1, **Accuracy** measures the fraction of agent–round responses for which the predicted choice matches the observed choice. **Macro F1** computes F1 separately for each response class and averages the resulting class-wise scores, giving equal weight to common and rare responses. **Population KL** measures the divergence between predicted and observed population-level choice distributions. Together, these metrics evaluate response prediction at both the individual and population levels. For RQ2, **Success** measures the fraction of agent–rounds successfully served, while **Crowd-out** measures failures caused by exhausted capacity when multiple agents select the same option. Success and Crowd-out are reported as proportions in $[0, 1]$ and rounded to three decimals. Exact formulas are given in Appendix F.

5.2 RQ1: Response prediction

Table 1 evaluates response prediction on held-out contexts and allocations covered by randomized collection. We compare the full model with the current-context, baseline-SFT, no-IB history-encoder, MF-LLM, and retrieval variants listed in the table.

Result. BRIA performs best on all three response metrics across all three datasets, indicating that it improves both individual response prediction and the recovery of population-level choice patterns. The consistent advantage over current-context, no-IB, MF-LLM, and retrieval baselines suggests that accurate allocation-conditioned response modeling requires more than fitting the latest observation or retrieving relevant history: the model must compress interaction history into a representation that preserves the behavioral information needed to predict how agents respond to different joint information allocations.

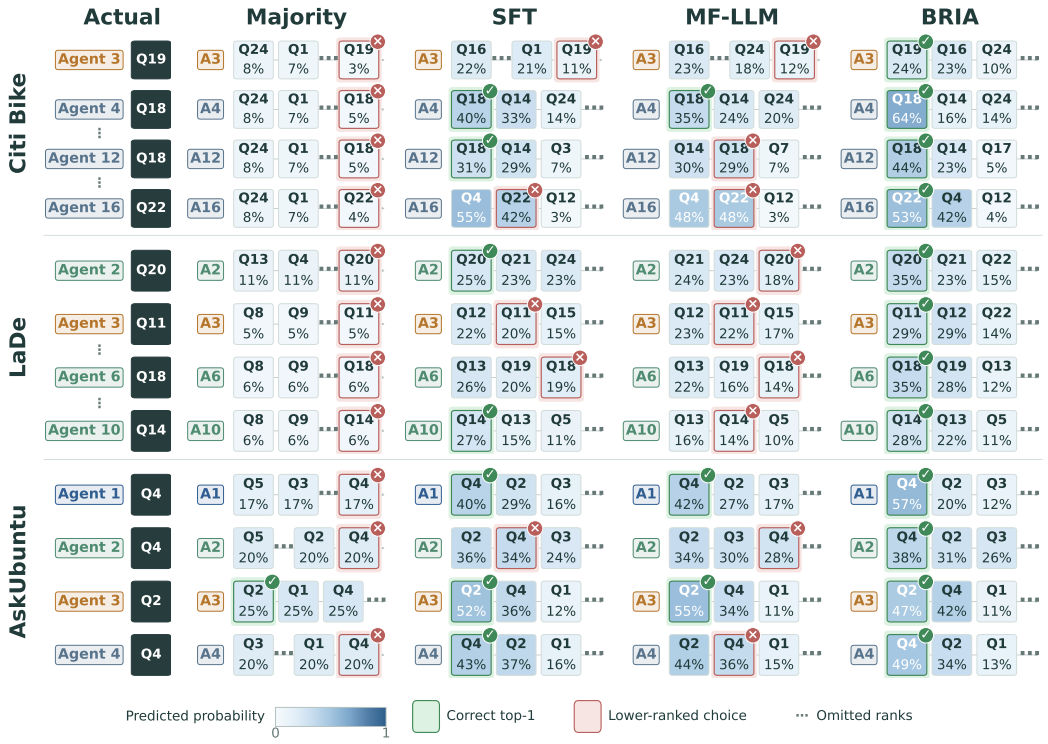


Figure 3: **RQ1 response prediction.** Representative held-out predictions; green marks a correct top-1 response and red marks the realized response when ranked lower.

Table 1: **Response prediction on held-out allocation settings covered by randomized collection.** Higher accuracy and macro F1 are better, while lower KL is better.

Response model	Citi Bike			LaDe			AskUbuntu		
	Acc.↑	MaF1↑	KL↓	Acc.↑	MaF1↑	KL↓	Acc.↑	MaF1↑	KL↓
Most-frequent response	0.097	0.023	0.721	0.499	0.142	0.790	0.324	0.232	0.946
Current context only	0.478	0.469	0.597	0.728	0.476	0.481	<u>0.732</u>	<u>0.650</u>	0.370
Baseline SFT	0.468	0.458	0.609	0.732	0.482	0.471	0.728	0.642	0.377
History encoder w/o IB	0.463	0.451	0.612	0.736	<u>0.491</u>	0.466	0.730	0.642	0.374
MF-LLM	0.465	0.457	0.610	0.718	0.461	0.501	0.728	0.647	<u>0.369</u>
Retrieval	<u>0.483</u>	<u>0.472</u>	<u>0.593</u>	<u>0.736</u>	0.487	<u>0.465</u>	0.718	0.629	0.384
BRIA response model	**0.526**	**0.504**	**0.535**	**0.756**	**0.523**	**0.436**	**0.742**	**0.662**	**0.364**

5.3 RQ2: End-to-end allocation

Table 2 reports outcomes from direct execution. Base context and Full-information test the effect of information volume, while all allocation methods use the same per-agent fact budget. CoT and ReAct are separate controls because they modify acting-agent inference rather than allocation.

Result. BRIA achieves the highest Success and the lowest Crowd-out on all three datasets, showing that its gains come not only from serving more agents but also from reducing harmful concentration on scarce options. Full-information helps on AskUbuntu but hurts on Citi Bike and LaDe, indicating that coordination depends on how information is distributed rather than on information volume alone. Likewise, per-agent relevance and diversity methods remain below BRIA, suggesting that independently useful information does not account for population-level response externalities. The strongest learned baselines narrow the Success gap but still leave more crowding. On LaDe, for example, Success changes only from 0.421 to 0.422 relative to Joint REINFORCE, while Crowd-out drops from 0.260 to 0.225. This highlights BRIA's main advantage: response-guided joint allocation

Table 2: **RQ2: End-to-end allocation results.** Higher Success and lower Crowd-out are better.

Method	Citi Bike		LaDe		AskUbuntu	
	Success↑	Crowd-out↓	Success↑	Crowd-out↓	Success↑	Crowd-out↓
Information controls						
Base context	0.441	0.441	0.401	0.300	0.436	0.453
Full-information	0.324	0.617	0.395	0.309	0.475	0.415
Non-learned allocation baselines						
Uniform random	0.453	0.430	0.411	0.280	0.475	0.410
Per-agent retrieval	0.403	0.464	0.408	0.287	0.490	0.390
MMR retrieval	0.450	0.439	0.413	0.276	0.490	0.388
DPP retrieval	0.463	0.426	0.407	0.282	0.492	0.389
Context compression	0.467	0.425	0.403	0.291	0.482	0.399
Recent-peer	0.277	0.530	0.409	0.290	0.482	0.401
Popularity-peer	0.283	0.528	0.401	0.282	0.480	0.402
Learned allocation baselines						
Joint REINFORCE	0.501	0.400	<u>0.421</u>	<u>0.260</u>	0.502	0.376
MAPPO	<u>0.503</u>	<u>0.394</u>	0.411	0.270	<u>0.503</u>	<u>0.374</u>
QMIX	0.459	0.443	0.408	0.279	0.477	0.403
Stronger-agent controls (no allocation)						
CoT (base context)	0.438	0.440	0.407	0.290	0.438	0.454
ReAct (base context)	0.418	0.465	0.399	0.299	0.480	0.408
BRIA	**0.546**	**0.363**	**0.422**	**0.225**	**0.530**	**0.340**

can steer agents away from mutually interfering choices, whereas stronger individual reasoning with CoT or ReAct does not consistently remove the same coordination failure.

5.4 Ablation Analysis

Table 3 removes one component at a time on Citi Bike. The inner-level variants test history, the information bottleneck, and response coupling; the outer-level variants test joint proposal and repeated response-model refitting.

Table 3: **Ablations on Citi Bike.**

Variant	Succ.↑	Cr.-out↓
w/o History	0.541	<u>0.367</u>
w/o IB	0.537	0.371
w/o Response Coupling	<u>0.542</u>	0.377
w/o Joint Proposal	0.535	0.374
w/o Refitting	0.535	0.371
w/o LM Fine-tuning	0.531	0.380
w/o Outcome Scoring	0.521	0.377
BRIA	**0.546**	**0.363**

Result. The ablations reveal a clear hierarchy among the design choices. Removing outcome scoring causes the largest Success drop, showing the importance of evaluating allocations by their predicted collective consequences. Removing the joint proposal or later response-model refitting also degrades performance, indicating that coordination benefits from both population-aware allocation and continual adaptation to the histories induced by the allocator. History modeling, information bottlenecking, response coupling, and auxiliary-model fine-tuning provide additional gains. Overall, the results support BRIA as a closed-loop system in which response modeling and joint allocation reinforce each other.

6 Conclusion

We studied how coordination in LLM-agent populations can be improved by controlling who sees what without modifying the acting agents themselves. BRIA combines a history-aware population response model with joint information allocation in a closed loop, improving response prediction and end-to-end coordination while reducing crowd-out across three real-world datasets. **Limitations.** Our study focuses on settings with selectively allocable truthful information and joint-outcome evaluation; future work can extend BRIA to richer interactions, longer horizons, and more heterogeneous agent populations.

Ethics statement

Our method changes which truthful optional state facts are shown to agents before they act. Although BRIA does not fabricate facts or alter feasible actions, selective disclosure can still influence behavior and can distribute benefits or burdens unevenly across agents. Aggregate improvements should therefore be evaluated together with per-agent outcomes, including whether particular agents are systematically given less useful information or experience worse task outcomes. In deployments involving people or consequential decisions, the allocation objective should incorporate application-specific fairness or minimum-service constraints, and allocations should be auditable so that systematic disparities can be detected. The allocator should also be restricted to information that the recipient is authorized to receive; truthful information is not necessarily appropriate to disclose merely because it is predictive or useful for coordination. Our experiments use public records and truthful task facts derived from those records, and are intended to study coordination effects rather than to justify unrestricted information steering in real-world decision systems.

Reproducibility statement

We provide the problem formulation, method, and experiments in Sections 3, 4, and 5, with additional experimental details in the appendix. Our anonymized code is publicly available at https://anonymous.4open.science/r/Bilevel_Information_Allocation-02DE, and the data used in our experiments are included in the supplementary materials.

AI use statement

AI tools were used only for language polishing and improving the clarity of the writing.

References

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8053–8072, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.448. URL https://aclanthology.org/2025.findings-naacl.448/.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HyxQzBceg.

Ariel Flint Ashery, Luca Maria Aiello, and Andrea Baronchelli. Emergent social conventions and collective bias in LLM populations. *Science Advances*, 11(20):eadu9368, 2025. doi: 10.1126/sciadv.adu9368. URL https://doi.org/10.1126/sciadv.adu9368.

Dirk Bergemann and Stephen Morris. Information design: A unified perspective. *Journal of Economic Literature*, 57(1):44–95, 2019. doi: 10.1257/jel.20181489. URL https://doi.org/10.1257/jel.20181489.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 335–336. Association for Computing Machinery, 1998. doi: 10.1145/290941.291025. URL https://doi.org/10.1145/290941.291025.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, Doha, Qatar, 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Citi Bike. Citi bike system data. Online data portal, 2026. URL https://citibikenyc.com/system-data. Accessed 2026-09-25.

Svenja M. Griesbach, Martin Hoefer, Max Klimm, and Tim Koglin. Information design for congestion games with unknown demand. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9722–9730, 2024. doi: 10.1609/aaai.v38i9.28830. URL https://ojs.aaai.org/index.php/AAAI/article/view/28830.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, Huan Sun, and Yu Su. Is your LLM secretly a world model of the internet? model-based planning for web agents. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=c6l7yA0HSq.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Emir Kamenica and Matthew Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6): 2590–2615, 2011. doi: 10.1257/aer.101.6.2590. URL https://doi.org/10.1257/aer.101.6.2590.

Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286, 2012. doi: 10.1561/2200000044. URL https://doi.org/10.1561/2200000044.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1279–1289, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1153. URL https://aclanthology.org/N16-1153/.

Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. Learning to model the world with language. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 29992–30017. PMLR, 2024a. URL https://proceedings.mlr.press/v235/lin24g.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, 2024b. URL https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Qirui Mi, Mengyue Yang, Xiangning Yu, Zhiyu Zhao, Cheng Deng, Bo An, Haifeng Zhang, Xu Chen, and Jun Wang. MF-LLM: Simulating population decision dynamics via a mean-field large language model framework. In *Advances in Neural Information Processing Systems*, volume 38, pp. 86523–86561, 2025. doi: 10.52202/085713-2610.

Yansong Ning, Jun Fang, Naiqiang Tan, and Hao Liu. Agent-Omit: Adaptive context omission for efficient LLM agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.04284.

Joon Sung Park, Joseph C. O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 2018. URL https://proceedings.mlr.press/v80/rashid18a.html.

Christoph Riedl. Emergent coordination in multi-agent language models. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.05174.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in LLM-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Lin Sun, Chuang Liu, Can Zhang, Yubin Wu, Weijia Lu, and Ning Wu. AgentPO: Enhancing multi-agent collaboration via reinforcement learning. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10011478.

Hao Tang, Darren Key, and Kevin Ellis. WorldCoder, a model-based LLM agent: Building world models by writing code and interacting with the environment. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2243. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/820c61a0cd419163ccbd2c33b268816e-Abstract-Conference.html.

Song Wang, Zhen Tan, Zihan Chen, Shuang Zhou, Tianlong Chen, and Jundong Li. AnyMAC: Cascading flexible multi-agent collaboration via next-agent prediction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11555–11567, Suzhou, China, 2025a. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.584. URL https://aclanthology.org/2025.emnlp-main.584/.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, Vienna, Austria, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.1170. URL https://aclanthology.org/2025.acl-long.1170/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Lixia Wu, Haomin Wen, Haoyuan Hu, Xiaowei Mao, Yutong Xia, Ergang Shan, Jianbin Zheng, Junhong Lou, Yuxuan Liang, Liuqing Yang, Roger Zimmermann, Youfang Lin, and Huaiyu Wan. LaDe: The first comprehensive last-mile express dataset from industry. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5991–6002. Association for Computing Machinery, 2024. doi: 10.1145/3637528.3671548. URL https://doi.org/10.1145/3637528.3671548.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1787. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9c1535a02f0ce079433344e14d910597-Abstract.html.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

Appendix Contents

A. Variational Information Bottleneck
B. Proposal Update
C. Coupled Optimization Structure
D. Stability of the Closed-Loop Alternating Updates
E. Coverage Example
F. Metric Definitions
G. Evaluation Protocol
H. Implementation Details
I. Baseline Definitions

A Variational Information Bottleneck

We derive the variational bounds used to pass from Eq. equation 7 to Eq. equation 8. The argument follows the standard variational information-bottleneck construction (Alemi et al., 2017); MF-LLM (Mi et al., 2025) uses the same fixed-prior device for compressing population history. BRIA differs in conditioning the bottleneck on the current state and the predictive term on the candidate allocation.

Let $C = \mathcal{C}$, and let expectations be under the randomized-data distribution with $M \sim q_\phi(M \mid C, X)$.

Compression bound. By definition,

$$I(M; C \mid X) = \mathbb{E} \log \frac{q_\phi(M \mid C, X)}{q_\phi(M \mid X)}. \tag{20}$$

For any $r_0(M \mid X)$ independent of C given X, add and subtract $\log r_0(M \mid X)$:

$$\begin{aligned} &\mathbb{E} D_{\mathrm{KL}}\left(q_\phi(M \mid C, X) \,\|\, r_0(M \mid X)\right) \\ &= I(M; C \mid X) + \mathbb{E}_X D_{\mathrm{KL}}\left(q_\phi(M \mid X) \,\|\, r_0(M \mid X)\right) \\ &\geq I(M; C \mid X). \end{aligned} \tag{21}$$

This gives the compression bound used in the main-text objective. The slack is exactly the divergence between the encoder marginal and the fixed prior. A prior conditioned on the full history could track the encoder and collapse this regularizer, which is why r_0 is not allowed to depend on C beyond X.

Prediction bound. Using conditional mutual information,

$$I(M; A \mid X, K) = H(A \mid X, K) - H(A \mid X, K, M). \tag{22}$$

For any normalized predictor $\widehat{p}_\psi$,

$$H(A \mid X, K, M) \leq -\mathbb{E} \log \widehat{p}_\psi(A \mid X, M, K), \tag{23}$$

because cross-entropy upper-bounds conditional entropy. Therefore

$$I(M; A \mid X, K) \geq H(A \mid X, K) + \mathbb{E} \log \widehat{p}_\psi(A \mid X, M, K), \tag{24}$$

which gives the prediction bound used in the main-text objective.

Multiplying Eq. equation 21 by β_{IB}, subtracting Eq. equation 24, and dropping the term independent of (ϕ, ψ) yields Eq. equation 8.

Autoregressive response factorization. For $Z = (X, M, K_{1:n})$, the probability chain rule gives

$$p(A_{1:n} \mid Z) = p(A_1 \mid Z) \prod_{i=2}^{n} p(A_i \mid Z, A_{<i}). \tag{25}$$

Thus Eq. equation 6 is a general factorization of a joint response distribution under a chosen ordering; it does not impose conditional independence across current agent responses.

B Proposal Update

This appendix derives the target proposal and its finite-candidate approximation.

Reference policy. The reference policy has full support over valid allocations. A simple implementation samples one valid κ-fact subset uniformly for each agent:

$$g_0(K \mid X) = \prod_{i=1}^{n} \binom{|\mathcal{F}_i(X)|}{\kappa}^{-1}, \qquad K \in \mathcal{K}_t(X). \tag{26}$$

Proof of Proposition 1. Fix the current context and abbreviate $Q(k) = \overline{Q}_t(k \mid m)$. Equation equation 15 becomes

$$\min_g \sum_k g(k) Q(k) + \tau \sum_k g(k) \log \frac{g(k)}{g_0(k)} \quad \text{s.t.} \quad \sum_k g(k) = 1. \tag{27}$$

The Lagrangian stationarity condition gives

$$Q(k) + \tau \left(\log \frac{g(k)}{g_0(k)} + 1 \right) + \nu = 0, \tag{28}$$

so $g(k) = c\, g_0(k) e^{-Q(k)/\tau}$. Normalizing over the finite valid allocation set gives $c = 1/Z$ with Z defined in Eq. equation 16. Since $\tau > 0$ and g_0 has full support, the objective is strictly convex in g, giving a unique optimum.

Finite candidate approximation. Let g_{samp} denote the distribution used to draw a candidate batch. It mixes the current proposal snapshot with the reference policy:

$$g_{\text{samp}}(k) = (1-\rho)g_{\theta_{\text{old}}}(k) + \rho g_0(k). \tag{29}$$

The proposal snapshot θ_{old} is held fixed while the batch is scored. For the ideal target,

$$\frac{g^*(k)}{g_{\text{samp}}(k)} \propto \frac{g_0(k)}{g_{\text{samp}}(k)} e^{-Q(k)/\tau}. \tag{30}$$

Motivated by this ratio, we replace $Q(k)$ with the finite-rollout estimate $\widehat{Q}_b$ and normalize over the sampled candidates:

$$w_b = \frac{\frac{g_0(K^{(b)})}{g_{\text{samp}}(K^{(b)})} e^{-\widehat{Q}_b/\tau}}{\sum_{b'} \frac{g_0(K^{(b')})}{g_{\text{samp}}(K^{(b')})} e^{-\widehat{Q}_{b'}/\tau}}. \tag{31}$$

Substituting these weights into weighted maximum likelihood yields Eq. equation 17.

Bounded sampling correction. Because

$$g_{\text{samp}}(k) = (1-\rho)g_{\theta_{\text{old}}}(k) + \rho g_0(k) \geq \rho g_0(k),$$

the importance correction is bounded:

$$0 < \frac{g_0(k)}{g_{\text{samp}}(k)} \leq \frac{1}{\rho}. \tag{32}$$

Thus the reference-policy correction cannot become arbitrarily large. The finite-candidate update is a practical approximation motivated by the ideal target in Eq. equation 16, rather than an exact sample from it. The self-normalized estimator is generally biased at finite N_{cand}. In addition, because the exponential transform is nonlinear, replacing $Q(k)$ by the finite-rollout estimate $\widehat{Q}_b$ generally introduces bias in the exponential weights even when $\widehat{Q}_b$ is an unbiased estimator of $Q(k)$. This rollout-induced discrepancy vanishes as the rollout estimates concentrate around $Q(k)$.

C Coupled Optimization Structure

BRIA consists of two coupled optimization levels: a response-learning level that fits $\lambda = (\phi, \psi)$ from randomized response data, and an allocation level that updates θ using scores produced by the fitted response model. At each outer iteration, the response model is first fitted and then treated as fixed while the allocation proposal is updated, as summarized in Eqs. equation 18–equation 19.

The coupling operates in both directions. The fitted response parameters λ_u^* determine the model-based scores used to update θ_{u+1}, while the resulting allocator changes the interaction histories subsequently reached by the population. Randomized g_0 rounds collected on those histories then change the dataset used to fit the next response model. Thus, the two levels are linked through both model-based allocation scores and the history distribution induced by the evolving allocator.

This alternating structure differs from classical differentiable bilevel programming: BRIA does not require an exact inner optimum or differentiate the allocation objective through the optimizer used to fit the response model. The term *bilevel* therefore refers to the coupled two-level optimization structure used by the method.

D Stability of the Closed-Loop Alternating Updates

BRIA updates the response model and allocation proposal alternately rather than simultaneously. Within each update cycle, the response model is fitted and then held fixed while the proposal is updated; after the allocator changes, new randomized g_0 rounds refresh the response model on the newly reached histories.

Stability considerations. This alternating schedule prevents the two learned components from changing at the same time, while randomized g_0 rounds preserve broad allocation coverage as the allocator evolves. Together, these choices separate the main sources of non-stationarity and maintain randomized response supervision across outer updates.

E Coverage Example

Suppose there are $n \geq 2$ agents and n options, and each agent independently receives a uniformly sampled option. Let N_{cov} be the number of distinct options covered. For option j, define $I_j = 1$ if at least one agent is assigned j. Then

$$N_{\text{cov}} = \sum_{j=1}^{n} I_j, \qquad \mathbb{E}[N_{\text{cov}}] = \sum_{j=1}^{n} \Pr(I_j = 1). \tag{33}$$

Under independent uniform assignment,

$$\Pr(I_j = 0) = \left(1 - \frac{1}{n}\right)^n, \tag{34}$$

so

$$\mathbb{E}[N_{\text{cov}}] = n\left[1 - \left(1 - \frac{1}{n}\right)^n\right] < n, \tag{35}$$

which gives Eq. equation 11. Full coverage occurs only when all n assignments are distinct:

$$\Pr(N_{\text{cov}} = n) = \frac{n!}{n^n}. \tag{36}$$

Now sample a uniformly random permutation of the n options and assign one option to each agent. Every agent still has a uniform marginal over options, but every option appears exactly once, so

$$\Pr(N_{\text{cov}} = n) = 1. \tag{37}$$

Hence identical per-agent marginals can yield different population coverage depending on the dependence structure of the joint assignment.

F Metric Definitions

Let $t = 1, \ldots, T$ index evaluation rounds, let n_t be the number of agents in round t, and let $N = \sum_t n_t$. For RQ1, y_{ti} is the observed response of agent i, $\hat{y}_{ti}$ is the most probable predicted response, and $p_{ti}(a)$ is the predicted marginal probability of response a. A no-response outcome is treated as its own response class.

For the autoregressive population model, all reported prediction probabilities are obtained without conditioning on ground-truth contemporaneous responses $A_{<i}$. Agent-level marginals are estimated from the model's joint predictive distribution using ancestral sampling.

Accuracy.

$$\text{Acc} = \frac{1}{N} \sum_{t=1}^{T} \sum_{i=1}^{n_t} \mathbf{1}[\hat{y}_{ti} = y_{ti}]. \tag{38}$$

Macro F1. Let $\mathcal{Y}$ be the response classes that occur in the evaluation targets. For class c, let TP_c be the number of correct predictions of c, and let N_c^{true} and N_c^{pred} be its true and predicted counts. Then

$$\text{MaF1} = \frac{1}{|\mathcal{Y}|} \sum_{c \in \mathcal{Y}} \frac{2TP_c}{N_c^{\text{true}} + N_c^{\text{pred}}}. \tag{39}$$

Population KL. Let $\mathcal{A}_t$ contain the response options available in round t together with the no-response class. Define the observed and predicted population shares

$$P_t(a) = \frac{1}{n_t} \sum_i \mathbf{1}[y_{ti} = a], \qquad Q_t(a) = \frac{1}{n_t} \sum_i p_{ti}(a). \tag{40}$$

For numerical stability, each entry of Q_t is lower-bounded by $\epsilon = 10^{-12}$ and the vector is renormalized to obtain $\widetilde{Q}_t$. We report

$$\text{KL} = \frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{a \in \mathcal{A}_t \\ P_t(a) > 0}} P_t(a) \log \frac{P_t(a)}{\widetilde{Q}_t(a)}. \tag{41}$$

Success. Let $Y_{ti} = 1$ if agent i is successfully served in round t and 0 otherwise. Agents with no eligible action remain in the denominator and count as unsuccessful:

$$\text{Success} = \frac{\sum_{t,i} Y_{ti}}{\sum_t n_t}. \tag{42}$$

Experimental tables report Success as a proportion in $[0, 1]$, rounded to three decimals.

Crowd-out. Let L_t be the number of agents in round t that are not served because aggregate demand for their selected option exceeds that option's capacity available at the start of the round, and let A_t be the number of agents with at least one eligible action. Then

$$\text{Crowd-out} = \frac{\sum_t L_t}{\sum_t A_t}. \tag{43}$$

Failures caused by having no eligible action or selecting an option with zero capacity at the start of the round are excluded from L_t. If $\sum_t A_t = 0$, Crowd-out is defined as 0. Experimental tables report Crowd-out as a proportion in $[0, 1]$, rounded to three decimals.

G Evaluation Protocol

Each dataset is divided chronologically into profile construction, initial randomized collection, online adaptation, and final testing. Evaluation episodes use matched initial states and held-out exogenous sequences across methods. Subsequent states evolve from the realized joint actions, so different allocation methods may visit different endogenous states.

The collection branch is logged at every round. Only rounds generated by g_0 enter $\mathcal{D}^{\text{exp}}$ for response-model refitting. Selector rounds still affect later histories, so the randomized data are collected on histories induced by the evolving selector.

All allocation methods that select optional facts use the same action identities, capacities, eligibility rules, tie-breaking rule, agent profiles, memory initialization, task transitions, acting-agent prompts, decoding settings, and number of facts per agent. Parsed task actions form the response vocabulary; prediction categories follow Appendix F, including the no-response class.

H Implementation Details

Trainable components. The auxiliary history encoder q_ϕ, response model $\widehat{p}_\psi$, and allocation proposal g_θ are trained by BRIA. The acting-agent model is not fine-tuned by the allocation method. The history encoder conditions the response model and allocator through the latent variable M_t in Eq. equation 4; the context-only prior $r_0(M \mid X)$ is held fixed when optimizing Eq. equation 8.

Representative LaDe compute environment. The training node is equipped with 8 NVIDIA GeForce RTX 4090 GPUs (24 GiB each), 503 GiB system memory, and Ubuntu 22.04. Experiments use PyTorch 2.6.0+cu124 with CUDA 12.4.

RQ1: response-model configuration. The LaDe response-model experiment uses 14 agents and an initial randomized dataset D_0 of 600 valid rounds, split into 480 training rounds and 120 model-selection rounds (seed 0). The response model is Qwen2.5-0.5B-Instruct (Yang et al., 2024) with LoRA (Hu et al., 2022) rank 16, scaling factor 32, and dropout 0.05. The history encoder is a GRU (Cho et al., 2014) with hidden dimension 128, latent dimension 64, and history window 10. Optimization uses AdamW (Loshchilov & Hutter, 2019) with learning rate 3×10^{-4} for the response encoder and 1×10^{-5} for the LoRA parameters, information-bottleneck coefficient $\beta_{\text{IB}} = 0.0073$, batch size 6 rounds, and gradient accumulation over 2 batches. Model selection uses validation negative log-likelihood, with at most 20 epochs per training stage and early-stopping patience 5.

RQ2: allocation configuration. The acting agents use Qwen3-8B-AWQ (Yang et al., 2025; Lin et al., 2024b) with temperature 0.7, top-p 0.95, and maximum output length 1024 tokens. Allocation training performs 5 outer updates with 60 valid deployment rounds per update, for 300 valid rounds

in total (seed 0). The response model is initialized from the RQ1 model and refit after each outer update. Each agent receives $\kappa = 4$ facts. At each decision, the allocator evaluates $N_{\text{cand}} = 12$ candidates with $N_{\text{roll}} = 8$ model rollouts per candidate and planning horizon $T_{\text{p}} = 1$. The allocator learning rate is 10^{-4}, the Boltzmann temperature in Eq. equation 16 is $\tau = 0.1$, and the randomized collection mixture uses $\eta = 0.2$.

Rollout and allocation evaluation. At each decision point, one M_t is sampled and reused across all candidates so that candidate scores are compared under the same sampled history representation. All reported experiments use $T_{\text{p}} = 1$. For $T_{\text{p}} > 1$, the model value in Eq. equation 12 is defined under a specified continuation policy and state/history transition model; an implementation can realize this, for example, by re-encoding the simulated history or by learning a latent transition model.

For proposal sampling, admissible facts are indexed in a canonical order and each recipient's κ indices are generated autoregressively with masks that enforce increasing indices and leave enough remaining facts to complete the subset. Hence each subset has one generation path and its proposal likelihood is the product of the corresponding masked conditional probabilities. This makes both $g_\theta(K \mid X_t, M_t)$ and $g_0(K \mid X_t)$ exactly evaluable in Eq. equation 31.

RQ1 predictive evaluation. For the autoregressive response model, held-out prediction metrics are computed from the model's joint predictive distribution without feeding ground-truth contemporaneous responses $A_{<i}$ at evaluation time. Marginal per-agent response probabilities are estimated by ancestral sampling and are then used to compute Accuracy, Macro F1, and Population KL as defined in Appendix F.

Learned-baseline configuration. Joint REINFORCE (Williams, 1992), MAPPO (Yu et al., 2022), and QMIX (Rashid et al., 2018) use the same 5-by-60 outer training schedule and allocator learning rate 10^{-4}. MAPPO uses clipping coefficient 0.2, batch size 16 rounds, and 4 optimization epochs per update. QMIX uses hidden dimension 64 and $\epsilon = 0.1$. These baselines do not train the response-model inner loop used by BRIA.

I Baseline Definitions

This appendix states what each comparison in RQ1 and RQ2 does. Unless stated otherwise, allocation baselines use the same admissible fact family and give each agent κ facts, as in BRIA.

I.1 RQ1: response-prediction baselines

Most-frequent response. A non-parametric predictor that always outputs the most frequent response class in the training data. It uses no current state, allocation, or interaction history.

Current context only. A response predictor that conditions on the current public context and executed allocation but does not use the history representation M_t. It isolates the value of historical information beyond the current decision state.

Baseline SFT. A standard supervised fine-tuning baseline trained to predict observed agent responses from the available input context, without the VIB history objective used by BRIA. It measures whether ordinary response-model fine-tuning is sufficient.

History encoder without IB. This variant keeps the history encoder and response-prediction architecture but sets the information-bottleneck regularization to zero. It tests whether explicitly compressing history contributes beyond simply adding a learned history representation.

MF-LLM. A population-modeling baseline based on MF-LLM (Mi et al., 2025). It uses compressed population history to predict agent behavior, providing a comparison with prior population-level behavioral modeling.

Retrieval. A retrieval-based response baseline that supplies selected historical records directly as context instead of using the learned bottleneck representation M_t. It tests whether selecting relevant history can replace a learned history representation.

I.2 RQ2: END-TO-END ALLOCATION BASELINES

Base context. Agents receive only the mandatory base context and no optional state facts. This is a control for the amount of information shown rather than a direct allocation comparison.

Full-information. Every admissible optional state fact is shown to every eligible agent. This control tests whether simply showing every available fact improves coordination. Because it shows more optional information than the allocation methods, it serves as an information-amount control rather than a direct allocation comparison.

Uniform random. Each recipient receives a uniformly sampled valid κ-fact subset from its admissible fact set, corresponding to the full-support reference allocation policy g_0. It controls for the effect of receiving the same amount of truthful information without learned selection.

Per-agent retrieval. Each recipient independently receives its κ highest-scoring admissible facts under the same relevance scorer. Each recipient still receives κ facts, but selections are not coordinated across recipients.

MMR retrieval. Facts are selected independently for each recipient using maximal marginal relevance (MMR) (Carbonell & Goldstein, 1998), which balances relevance against redundancy within that recipient's selected set. It tests whether diversity within each agent's fact set is enough without coordinating across agents.

DPP retrieval. A determinantal-point-process (DPP) selector (Kulesza & Taskar, 2012) favors diverse fact subsets for each recipient while selecting κ facts. Like MMR, it diversifies each agent's fact set separately rather than optimizing the joint allocation.

Context compression. Each recipient's admissible information is compressed to κ facts conditioned on that recipient's current context. Agents are processed separately, so this tests whether context compression alone is enough without optimizing the joint population response.

Recent-peer. This heuristic conditions information selection on a summary of the most recent observed peer behavior. It tests whether recent peer behavior is enough without learning a population response model.

Popularity-peer. This heuristic uses aggregate peer-choice popularity as the population signal for information selection. It tests whether a simple popularity statistic can replace history-aware response modeling.

Joint REINFORCE. A direct policy-gradient baseline that learns the same joint allocation policy from realized task losses rather than from model-based candidate evaluation. For cost-to-go $G_t = \sum_{s=t}^{T} \ell_s$ and a learned baseline $V(X_t, M_t)$,

$$\widehat{\mathcal{L}}_{\text{direct}} = \frac{1}{|\mathcal{B}|} \sum_{(j,t)\in\mathcal{B}} \Big(G_t^{(j)} - V(X_t^{(j)}, M_t^{(j)})\Big) \cdot \log g_\theta\Big(K_{1:n}^{t,(j)} \mid X_t^{(j)}, M_t^{(j)}\Big). \tag{44}$$

The baseline is treated as a stop-gradient in the policy term. This comparison isolates the value of learning and reusing a response model instead of relying only on costly realized outcomes.

MAPPO. MAPPO (Yu et al., 2022) is adapted to learn the allocation policy under the same task interface, with κ facts per agent and the same outer training schedule. It provides a standard centralized-training multi-agent policy-gradient comparison without BRIA's response-model inner loop.

QMIX. QMIX (Rashid et al., 2018) is adapted to the same allocation environment and training schedule using value decomposition for the allocation decision. It provides a value-based multi-agent learning comparison without BRIA's learned response model.

CoT (base context). The acting agents use chain-of-thought prompting (Wei et al., 2022) while receiving only the base context, with no learned information allocator. This control tests whether stronger individual reasoning alone removes the observed coordination failures.

ReAct (base context). The acting agents use the ReAct prompting pattern (Yao et al., 2023) with the base context and no learned allocator. Like the CoT control, it changes the acting-agent inference procedure rather than the information-allocation policy.

I.3 Ablation definitions

w/o Interaction History. This ablation removes $\mathcal{C}_t$ from the history encoder, so the response model does not use interaction history.

w/o IB Regularization. This ablation sets $\beta_{\mathrm{IB}} = 0$ in Eq. equation 8 while keeping the history encoder and response architecture unchanged.

w/o Response Coupling. This inner-level ablation changes only the response model p_ψ. The full model uses Eq. equation 6, so the prediction for agent i can condition on the earlier predicted choices $A_{<i}$. The ablation removes this conditioning:

$$\widehat{p}_\psi^{\,\mathrm{ind}}(A_{1:n} \mid X, M, K) = \prod_{i=1}^{n} \widehat{p}_{\psi,i}(A_i \mid X, M, K). \tag{45}$$

Each agent's response is therefore predicted independently of the other agents' current choices.

w/o Joint Proposal. This outer-level ablation keeps the full response model p_ψ unchanged and changes only the allocation proposal g_θ. Equation equation 10 is replaced by

$$g_\theta^{\mathrm{ind}}(K_{1:n} \mid X, M) = \prod_{i=1}^{n} g_{\theta,i}(K_i \mid X, M). \tag{46}$$

Thus each agent's information set is proposed independently of what has been assigned to the other agents.

w/o Refitting. The response model is fitted once on the initial randomized data and then held fixed while the allocation proposal is trained. The later response-model refits on newly collected randomized data are removed; the allocator continues to be trained using the fixed response model.

w/o LM Fine-tuning. The auxiliary language model used by the response predictor is kept fixed while the remaining response-model components are trained normally.

w/o Outcome Scoring. Candidate allocations are not re-ranked by the response-model estimate $\widehat{Q}_b$ before execution.