Narrowing the Collaboration Gap, Probably

Anonymous Authors[1]

Abstract

Large language models are increasingly deployed as teams of agents that hold different private information about a shared task. How should such agents concisely communicate in a way that promotes aggregation of decision-relevant information? Communicating only proposed actions can discard decision-relevant uncertainty: two agents may prefer the same action for different reasons. We study an alternative interface in which agents communicate calibrated beliefs over actions. We first analyze a simple synthetic setting that captures the structure of multi-round reasoning under partial information. In this setting, we show theoretically and empirically that communicating unbiased probabilities can be strictly more powerful than communicating actions. We then test the same principle in a collaborative maze-solving task with trained transformers and pretrained LLMs. Our results demonstrate that probability communication outperforms action communication when the exchanged predictions preserve calibrated uncertainty. When communicated probabilities are biased, we show that the benefits of probability communication can disappear. We give a post-hoc conversation calibration intervention that consistently improves decision-making by correcting these biases. These results suggest that probabilistic communication is useful for multi-agent collaboration not merely because it transmits more information, but because calibrated uncertainty gives other agents a reliable object to update on.

1. Introduction

Large language models (LLMs) are increasingly used within multi-agent systems where agents may share a task but hold different private information. For example, physical robots may see different sensor data, and coding agents may inspect different files when the relevant repository is too large for any single context window. To succeed in such settings, agents must communicate to combine their private information to make better decisions than any single agent could alone.

In practice, multi-agent collaboration falls short of this goal because the same constraints that motivate multi-agent design often rule out the centralized solution of sending all private context to one agent. Communication may be limited by bandwidth, memory, context length, or by the fact that different agents' observations are not mutually legible across modalities. And even if one agent could transmit its full observation, the receiving agent may not be able to process it into a correct answer (Davidson et al., 2025). More generally, recent work has shown that LLMs often struggle in these collaborative settings, exhibiting failures in information sharing (Cemri et al., 2026), belief updating (Li et al., 2023; Wilie et al., 2024), and joint decision-making under partial information (Davidson et al., 2025). These failures reveal a broader "collaboration gap" (Davidson et al., 2025): current LLM agents do not reliably aggregate distributed observations into well-coordinated decisions. Narrowing this gap requires answering a fundamental question:

What should agents communicate in order to aggregate distributed private information?

A natural low-bandwidth alternative to communicating their hidden information is for agents to exchange only their proposed actions. However, communicating actions can omit decision-relevant uncertainty. A promising direction comes from the agreement literature, which shows that Bayesian reasoners with different observations can exchange beliefs to converge towards predictions that aggregate their information (Frongillo et al., 2023). More recent work weakens Bayesian rationality to calibration-style conditions, including *conversation calibration*, and studies when agreement would lead to optimal actions given the agents' combined observations (Collina et al., 2025; 2026; Kearns et al., 2026). This perspective suggests that reliable multi-agent LLM collaboration may require mechanisms that go beyond ad hoc message exchange: agents must be able to recognize their own uncertainty and interpret others' uncertainty.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

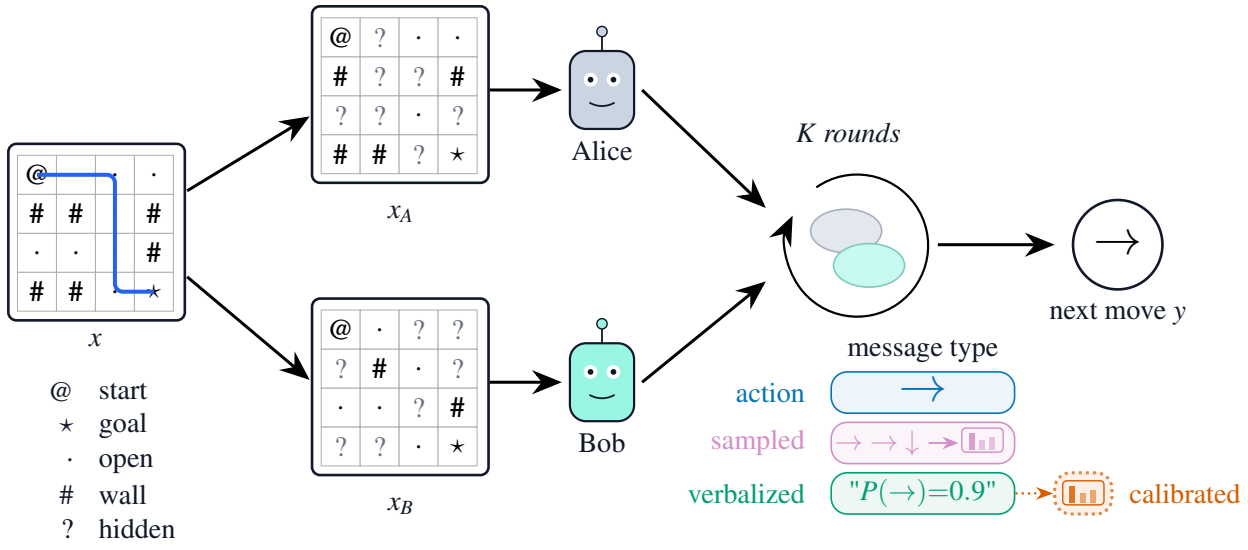

Figure 1. Multi-agent collaboration under partial information. Agents observe different views of the same maze and exchange messages over multiple rounds before producing a final action. We compare action-only messages, verbalized probabilities, sampled probabilities from multiple rollouts, and a post-hoc calibrated probability channel. *Figure adapted from (Davidson et al., 2025).*

1.1. Our contributions

Our hypothesis is that, under partial information, agents collaborate more effectively when they communicate calibrated beliefs rather than only their preferred decisions. We investigate this hypothesis theoretically in a stylized theoretical model and empirically on the collaborative maze task of Davidson et al. (2025). We study both transformers trained for the task and large-scale, pretrained LLM agents. Across these settings, we compare two low-bandwidth communication protocols:

1. *Action-only communication:* In each round, the speaker sends a single action that it currently believes is most likely correct. For transformers this is the action with highest softmax probability; for LLM agents, it is the action selected in the model's response.

2. *Probability communication:* In each round, the speaker sends a distribution over actions. For transformers this is the softmax vector. For LLM agents, we consider *verbalized probabilities*, where the model reports a probability vector in text, and *sampled probabilities*, we take the empirical distribution over independent response rollouts.

Our results are threefold. First, in a stylized chained-parity model of multi-round collaboration, we show a theoretical separation between communicating probabilities and communicating actions. Exact probability messages aggregate one new piece of information per round, ultimately leading to full information aggregation. Meanwhile, action-only messages can reveal only the first agent's information and then stall. Second, we test the same principle in controlled transformer experiments on collaborative maze solving (see Figure 1). We see that probability communication improves over action communication when the learned messages preserve calibrated uncertainty. When probability messages are *biased*, this advantage can disappear. To address this, we introduce a *post-hoc calibration algorithm* that corrects biased reports and recovers much of the performance lost when raw probability reports are distorted. Third, we evaluate the same communication protocols using pretrained LLM agents on the maze task. We find that simply asking models to verbalize probabilities is not enough: raw verbalized probabilities are often no better than action messages. We show that across the tested models, applying post-hoc calibration as an *intervention* consistently improves performance, indicating that probabilistic messages become substantially more useful when corrected into forecasts that other agents can reliably update on.

Beyond the agreement literature discussed above, our work also draws on research on calibrated forecasting, uncertainty elicitation in language models, sampling-based uncertainty, multi-agent LLM debate and collaboration, and learned communication in multi-agent reinforcement learning. We defer an extended discussion of these connections to Appendix A.

2. Preliminaries

Formally, we study a setting with two agents, Alice and Bob, who collaborate to find a correct action to take. Data points are drawn from a distribution over feature vectors

$x \in \mathcal{X}$, and each feature vector has a correct action $a^\star \in \mathcal{A}$. Neither agent observes the full feature vector: Alice observes $x_A \in \mathcal{X}_A$ and Bob observes $x_B \in \mathcal{X}_B$, with $x = (x_A, x_B) \in \mathcal{X}_A \times \mathcal{X}_B$. Thus the agents collectively have enough information to identify $a^\star$, but neither agent can do so reliably in isolation.

Communication protocol. The agents communicate for a fixed number of rounds K before selecting an action. Alice speaks on odd rounds and Bob on even rounds; we write $\sigma(k) \in \{A, B\}$ for the round-k speaker. We focus on the bandwidth-limited regime, in which each round's message is much smaller than the speaker's private observation.

We consider two types of communication protocols. In the *action-communication* protocol, the round-k message is a single proposed action $\hat{a}_k \in \mathcal{A}$. In the *probability-communication* protocol, the round-k message is a distribution over actions $\hat{p}_k \in \Delta(\mathcal{A})$. After round K, the agents execute the action chosen by the final speaker: the final action message $\hat{a}_K$ under action communication, and $\arg\max_a \hat{p}_K(a)$ under probability communication.

Calibration. Calibration formalizes when a probability can be interpreted as meaningful uncertainty. Intuitively, a probabilistic message is calibrated if its probabilities agree with long-run frequencies. For example, among all instances in which an agent assigns probability 0.7 to an action, that action should be correct around 70% of the time.

Definition 2.1 (Calibration)**.** Let $\hat{p}$ be a $\Delta(\mathcal{A})$-valued random variable, where the randomness is over the draw of the instance and any randomness in the agents' procedure. We say $\hat{p}$ is calibrated if, for every action $a \in \mathcal{A}$,$\Pr(a^\star = a \mid \hat{p}) = \hat{p}(a)$ almost surely.

Calibration is a marginal property: it says that an agent's stated uncertainty is correct in aggregate over all instances in which the agent sends the same distribution. With multiple agents, we may also ask if one agent's forecast remains calibrated after accounting for another communicated forecast. This is the notion of cross-calibration. Intuitively, Alice's forecast is cross-calibrated with Bob's forecast if, among all instances where Alice sends a particular distribution, subdividing those instances according to Bob's distribution does not change the correctness of Alice's probabilities.

Definition 2.2 (Cross-calibration)**.** Let $\hat{p}$ and $\hat{q}$ be $\Delta(\mathcal{A})$-valued random variables. We say $\hat{p}$ is cross-calibrated with respect to $\hat{q}$ if, for every action $a \in \mathcal{A}$, $\Pr(a^\star = a \mid \hat{p}, \hat{q}) = \hat{p}(a)$ almost surely.

Calibration requires an agent's probabilities to be correct in aggregate, while cross-calibration requires them to remain correct even after conditioning on another communicated forecast. This definition conditions only on the communicated probabilities, not on the other agent's private observation.

Definition 2.3 (Conversation calibration)**.** Following Collina et al. (2025), in a probability-communication protocol, the conversation is calibrated through round K if the initial forecast $\hat{p}_1$ is calibrated and, for every round $2 \le k \le K$, the round-k forecast is cross-calibrated with respect to the previous round's forecast: $\Pr(a^\star = a \mid \hat{p}_k, \hat{p}_{k-1}) = \hat{p}_k(a)$ for all $a \in \mathcal{A}$ almost surely.

3. When do probabilities outperform actions?

This section establishes when probability communication can be more useful than exchanging only actions. The gap appears when the best action is too coarse to reveal which evidence state produced it, which we demonstrate via a multi-round separation using chained parities. A warm-up example for one-round, where two posterior states induce the same best action but different probabilities, is included in Appendix B.

Chained-parity task. We consider a binary action space $\mathcal{A} = \{0, 1\}$. Fix $K \ge 1$ and write $d_A := \lceil K/2 \rceil$ and $d_B := \lfloor K/2 \rfloor$. Let $b_1, \ldots, b_K$ be independent uniform bits in $\{-1, +1\}$. Alice and Bob observe the odd- and even-indexed bits as fixed-length vectors:

$$x_A := (b_1, b_3, \ldots, b_{2d_A - 1}) \in \{-1, +1\}^{d_A},$$
$$x_B := (b_2, b_4, \ldots, b_{2d_B}) \in \{-1, +1\}^{d_B}.$$

Define the *prefix parity*, the ± 1-valued product (equivalently, the XOR) of the first j bits,

$$S_0 := 1, \qquad S_j := \prod_{\ell=1}^{j} b_\ell \quad \text{for } j = 1, \ldots, K.$$

Conditional on $x = (x_A, x_B)$, the target action $a^\star$ is a Bernoulli random variable with parameter

$$p^\star := \frac{1}{2} + \sum_{j=1}^{K} \alpha_j S_j,$$

which we call the *full-information posterior probability* of $a^\star = 1$. Throughout this section, $p^\star$ denotes this random variable, a deterministic function of the bit vector $b = (b_1, \ldots, b_K)$. The weights $\alpha_1, \ldots, \alpha_K$ are positive and satisfy $\sum_{j=1}^{K} \alpha_j < 1/2$, which keeps $p^\star \in (0, 1)$. Define $h^\star := \mathbf{1}\{p^\star \ge \frac{1}{2}\} \in \{0, 1\}$, the Bayes-optimal action under 0–1 loss. Because each term $\alpha_j S_j$ with $j \ge 2$ depends on *both* agents' bits (Figure 2), no agent can evaluate $p^\star$ alone.

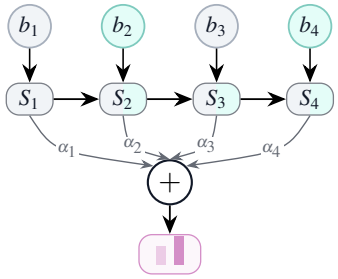

Figure 2. Chained-parity dependency structure.

Communication protocol. At round k, Alice speaks when k is odd and Bob speaks when k is even; write this speaker as $\sigma(k) \in \{A, B\}$. Thus the round-k speaker observes b_k. When Alice and Bob communicate probabilities, define $p_0 := 1/2$ and let the speaker's Bayes posterior probability that $a^\star = 1$, given their own features and the previous probability message, be

$$p_k := \Pr\left(a^\star = 1 \,\middle|\, x_{\sigma(k)}, p_{k-1}\right) \in [0, 1].$$

The probability communicated is $\hat{p}_k = p_k$. When actions are communicated instead, let $\tilde{p}_k$ be the analogous Bayes posterior computed from the speaker's own features and the public action transcript so far; the speaker's action that they communicate to the other player will then be their best response,

$$\hat{a}_k = \mathbf{1}\{\tilde{p}_k \geq \tfrac{1}{2}\} \in \{0, 1\}.$$

The distinction is that p_k can preserve how strongly the speaker favors an action, while $\hat{a}_k$ keeps only the sign of that preference.

Information aggregation with probabilities We now add the condition that makes the previous probability message a sufficient scalar summary of the parities revealed so far.

Assumption 3.1. Distinct bit prefixes $(S_1, \ldots, S_k) \in \{-1, +1\}^k$ yield distinct partial sums $\sum_{j \leq k} \alpha_j S_j$ for each $k \leq K$ (e.g. $\alpha_j = 2^{-j-1}$).

The distinct-partial-sums condition ensures that a scalar probability message identifies the prefix parities already revealed. Therefore the previous probability message is not merely a confidence score; it is an invertible summary of the information accumulated so far. This lets the probability exchange advance one message at a time:

$$p_k = p_{k-1} + \alpha_k S_k = \frac{1}{2} + \sum_{j \leq k} \alpha_j S_j.$$

By Assumption 3.1, the previous scalar message p_{k-1} identifies the prefix $(S_1, \ldots, S_{k-1})$; the round-k speaker also observes b_k, and hence determines $S_k = S_{k-1} b_k$. All later parity terms contain an unseen bit and have conditional mean zero. Since p_k is a Bayes posterior conditioned on the speaker's information and the previous probability message, the probabilities communicated satisfy conversation calibration in the sense of Definition 2.3.

Theorem 3.2 (Probability exchange aggregates information)**.** *In the chained-parity model, under Assumption 3.1, when probabilities are communicated, the round-k message equals $p_k = \frac{1}{2} + \sum_{j \leq k} \alpha_j S_j$ for every $k \leq K$; in particular $p_K = p^\star$. The round-k excess squared loss $\mathbb{E}[(a^\star - p_k)^2] - \mathbb{E}[(a^\star - p^\star)^2]$ equals $\sum_{j>k} \alpha_j^2$ and decreases by α_k^2 each round, so after K rounds the agents attain the full-information Bayes risk $\mathbb{E}[(a^\star - p^\star)^2]$.*

Proof sketch. The recurrence above gives $p_k = 1/2 + \sum_{j \leq k} \alpha_j S_j$ by induction, so $p_K = p^\star$. At any $k < K$, p_k differs from $p^\star$ by the unrevealed tail $\sum_{j>k} \alpha_j S_j$. Distinct parities are orthogonal, so the excess squared loss is $\mathbb{E}[(p^\star - p_k)^2] = \sum_{j>k} \alpha_j^2$, and the one-round improvement is α_k^2.

Information loss with actions. By contrast, the action exchange can collapse all later evidence states to the same first action.

Theorem 3.3 (Action exchange transmits only the first bit)**.** *In the chained-parity model, assume $K \geq 2$ and $\alpha_1 > \alpha_2$. When actions are exchanged, every round-k message equals $\mathbf{1}\{b_1 = +1\}$, so the induced decision is constant in k. In particular, for $K = 3$ and $\alpha = (0.20, 0.15, 0.10)$, this decision disagrees with the full-information Bayes action $h^\star$ on an event of probability $1/4$.*

Proof sketch. The action message keeps only the sign of the speaker's current posterior and discards the magnitude. Alice's first posterior is $1/2 + \alpha_1 b_1$, so her action message is $\mathbf{1}\{b_1 = +1\}$. If Alice speaks later, she still lacks b_2, so her posterior again has sign b_1. If Bob speaks, he can use b_2, but later terms average out and his posterior is $1/2 + \alpha_1 b_1 + \alpha_2 b_1 b_2 = 1/2 + b_1(\alpha_1 + \alpha_2 b_2)$. Since $\alpha_1 > \alpha_2$, this posterior also has sign b_1. Thus every action message repeats the first bit and no later bit is transmitted. For the displayed $K = 3$ coefficients, on the event $\{b_2 = -1, b_3 = +1\}$, the full posterior satisfies $p^\star - 1/2 = b_1(\alpha_1 - \alpha_2 - \alpha_3) < 0$, so $h^\star$ is the opposite of the action-channel decision; this event has probability $1/4$.

Empirical validation. We validate this separation with trained transformers on the same chained-parity task. Figure 3 shows that learned probability messages track the theoretical loss curve, while learned action messages stall as predicted (Appendix E for details).

4. When do probability reports need calibration?

Theorem 3.2 assumes the round-k speaker transmits its Bayes posterior p_k exactly. In practice, reported probabilities may be distorted . One way to model this failure

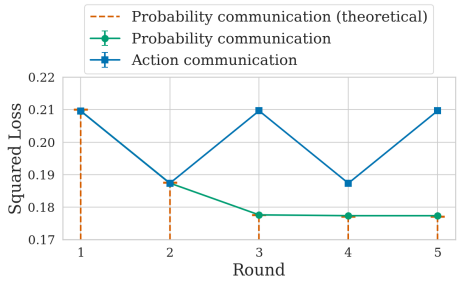

Figure 3. Squared loss of action vs probability exchange over $K = 5$ rounds on chained parities. Learned probability communication attains the theoretical optimum.

is to combine two deviations that are well-documented in LLMs: global miscalibration of confidence (Zhao et al., 2021; Xiong et al., 2024), where stated probabilities are systematically too extreme or too mild, and *sycophancy* (Perez et al., 2023; Sharma et al., 2023), where an agent's report is pulled toward the previous speaker's report regardless of its own evidence. We analyze a single round of such distortion: the previous agent is calibrated and sends the ideal p_{k-1}, while the round-k agent outputs a biased report,

$$\hat{p}_k = \tfrac{1}{2} + c\big(p_k - \tfrac{1}{2}\big) + \lambda\big(p_{k-1} - \tfrac{1}{2}\big), \tag{1}$$

with (c, λ) chosen so that $\hat{p}_k \in [0, 1]$ almost surely. The parameter $c > 0$ controls the confidence scale ($c = 1$ is calibrated, $c > 1$ overconfident, $c < 1$ underconfident), and $\lambda \in \mathbb{R}$ controls the strength of sycophantic anchoring to the previous report. This illustrates the broader problem: raw probability reports need not be reliable objects for the next agent to update on. To recover calibrated messages, we can learn the conditional label distribution from the communicated probabilities themselves.

Theorem 4.1 (Recovering guarantees via post-hoc calibration)**.** *In the chained parity model, suppose the previous message is the ideal posterior p_{k-1}. For any $c > 0$ and $\lambda \in \mathbb{R}$, let $Z = (\hat{p}_k, p_{k-1})$. If the predictor class contains the conditional label distribution, then training a predictor h to predict $a^\star$ from Z using a strictly proper loss yields, at the population optimum, the one-step post-hoc forecast $h^\star(\hat{p}_k, p_{k-1}) := \Pr(a^\star = 1 \mid \hat{p}_k, p_{k-1})$ which recovers the posterior: $h^\star(\hat{p}_k, p_{k-1}) = p_k$. Moreover, the learned forecast is cross-calibrated with respect to the previous one. Applied inductively, with each previous calibrated message equal to the corresponding ideal posterior, this recovers the guarantee of Theorem 3.2 for any (c, λ).*

Proof sketch. A strictly proper loss is minimized at the true conditional label distribution, so the population optimum satisfies $h^\star(Z) = \Pr(a^\star = 1 \mid Z)$. Since $c > 0$, observing $(\hat{p}_k, p_{k-1})$ in (1) determines p_k; the unrevealed tail still has conditional mean zero given these messages. Therefore $\Pr(a^\star = 1 \mid \hat{p}_k, p_{k-1}) = p_k$. Finally, because Z includes the previous forecast, the tower property gives $\Pr(a^\star = 1 \mid h^\star(Z), p_{k-1}) = h^\star(Z)$, so the learned forecast is cross-calibrated with respect to p_{k-1}. Full details are in Appendix D.

The affine form in (1) is only a convenient example. The same conclusion holds for any distortion $\hat{p}_k = F(p_k, p_{k-1})$ such that, after fixing p_{k-1}, the map $p_k \mapsto F(p_k, p_{k-1})$ is invertible on the possible round-k posteriors. The gap from calibration is already visible in the first two rounds: if a receiver treats a biased report as a valid posterior and projects it to the nearest possible message, the first bit can be decoded correctly while the second bit may be flipped by confidence or anchoring bias. Appendix D formalizes this.

Algorithm 1 Post hoc calibration for probability communication

Require: Calibration data $\{(x_i^A, x_i^B, a_i^\star)\}_{i=1}^n$, rounds K, trained base protocol producing raw forecasts $\tilde{p}^k$, strictly proper loss ℓ

Initialize calibrated previous forecast $\hat{p}^0 = \emptyset$
for $k = 1, \ldots, K$ **do**
  Let $s = \sigma(k)$ be the round-k speaker and let $q = \hat{p}^{k-1}$ be the previous calibrated forecast
  Run the base protocol to obtain the raw round-k forecast $\tilde{p}_i^k = \tilde{\pi}_s^k(x_i^s, q)$ for $i = 1, \ldots, n$
  Fit a post hoc calibration map using only the raw forecast and previous calibrated forecast:

$$g_k \in \arg\min_{g \in \mathcal{G}_k} \sum_{i=1}^{n} \ell\big(g(\tilde{p}_i^k, q_i), a_i^\star\big)$$

  Set the calibrated message $\hat{p}_i^k = g_k(\tilde{p}_i^k, q_i)$
end for
return Calibrators $g_1, \ldots, g_K$ =0

A post hoc calibration algorithm. The above observation suggests a natural training procedure for collaborative agents that communicate probabilistic predictions. In each round, the active agent produces a raw forecast from its own view and the previous calibrated message. A post-hoc map then recalibrates this forecast using labelled transcripts and the previous message as inputs. This yields an iterative procedure in which agents alternately improve their predictive accuracy while aligning their communicated probabilities through cross-calibration. This procedure is in Algorithm 1.

5. Maze experiments with learned and pretrained agents

We move beyond chained parities to test these communication principles on collaborative maze-solving. The task, proposed by Davidson et al. (2025) to test collaborative

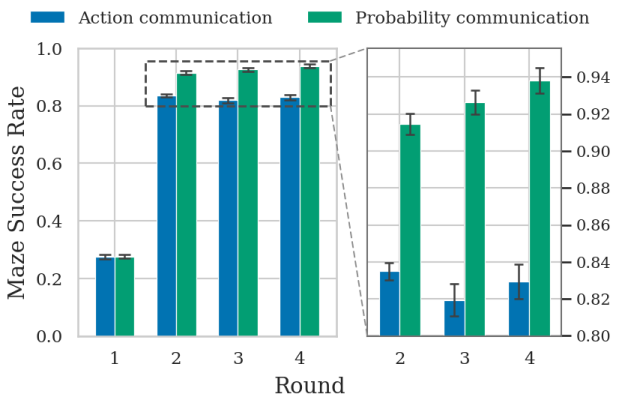

Figure 4. **(R1)** Action performance stagnates; communicating probabilities increases performance gradually.

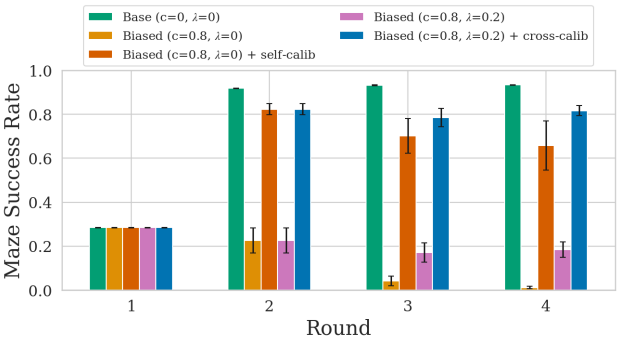

Figure 5. **(R2)** Global miscalibration and sycophantic anchoring cause performance to drop over rounds, while post-hoc recalibration partially recovers performance under self-miscalibration and cross-calibration recovers nearly all performance when both biases are present.

capabilities of language models, preserves the same partial-information structure in a more natural decision problem. We first study it in a controlled setup, where, as in Section 3, we train a sequence of transformers that communicate across rounds.

Maze task. The maze task consists of a dataset of $N \times N$ mazes, each with a fixed start state, fixed end state, and cells denoting either a path or a wall, see Figure 1. Mazes are sampled according to a wall-density parameter, which controls the expected fraction of cells that are walls. In our collaborative setup, each agent observes only a partial map of the maze with half of the cells hidden. Together, the two agents' maps recover the full maze. In our experiments, we use $N = 6$ and a wall density of 30%, matching the primary setting studied by Davidson et al. (2025). Unlike Davidson et al. (2025), we restrict attention to mazes with a single correct solution for ease of evaluation. An example maze (and the corresponding two views) are presented in Figure 9.

5.1. Controlled transformer experiments

Transformer setup. We train small transformers on a supervised next-move prediction task. Each example is represented as a partial-view maze string with the @ symbol denoting the current position, along with any previously communicated message. The target label is the next move along the solution path. Full experimental details are in Appendix E. We report maze success rate, defined as the fraction of mazes for which the model predicts every move in the solution path correctly.

(R1) Probability messages improve controlled maze-solving. Figure 4 shows that trained models under probability communication indeed outperform trained models under action communication. Moreover, it shows that over four rounds of communication, maze success rate improves consistently when agents communicate probabilities; this is not the case for action communication.

(R2) Calibration repairs biased transformer messages. Next, we study the two biases that cross-calibration is designed to mitigate when agents communicate probabilities: global miscalibration and sycophantic anchoring. Rather than communicating the transformer probabilities directly, we first distort them according to Equation 1, and then optionally apply a post-hoc debiasing protocol. The debiasing model is a single-layer network trained with cross-entropy loss against the true label. We compare self-calibration, which uses only the current transformer's probabilities, with conversation calibration, which uses both the current and previous round's probabilities.

The unbiased probability baseline improves monotonically over four rounds; see Figure 5. In contrast, global miscalibration alone and global miscalibration combined with anchoring cause performance to collapse and deteriorate over rounds. Self-calibration recovers a substantial fraction of the lost success under global miscalibration, but its imperfect corrections still accumulate through the conversation. When both biases are present, conversation calibration recovers nearly all of the baseline performance. Thus, cross-calibrating against the previous round's probabilities can recover approximately unbiased messages and substantially improve downstream maze success.

5.2. Pretrained LLM experiments

We now test if the same principle holds for large-scale, pretrained language models on the maze task.

LLM setup. We use the maze task described in Section 5 and evaluate four models of varying strengths: GPT-4.1-mini, GPT-5, GPT-5.4-nano, and GPT-5.4-mini. We allow models to generate reasoning (up to 2048 tokens). We consider three communication protocols: actions, verbalized probabilities, or sampled probabilities. In the *action*

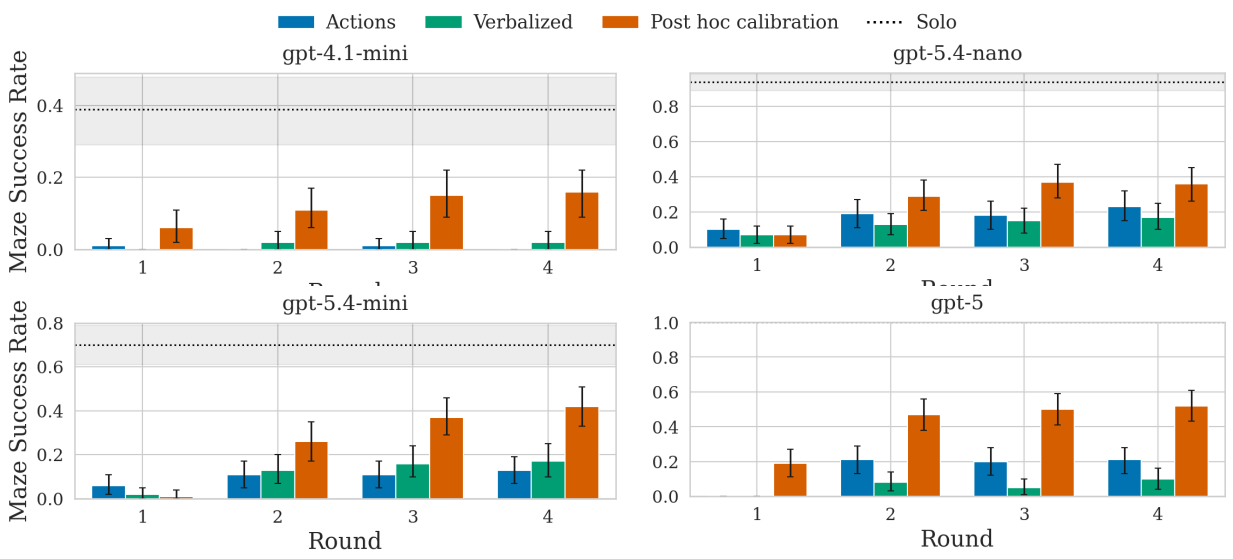

Figure 6. **(R3)** We train one-hidden-layer MLP calibrators according to Algorithm 1 on a set of 100 mazes; we report outcomes on a validation set of 100 mazes. We also include the full-information solo baseline. Bars that do not appear on the plot indicate zero success

exchange protocol, each agent communicates the action chosen in its response. In the *verbalized probability* exchange protocol, each agent communicates the probability vector reported in its response. In the *sampled probability* exchange protocol, we sample K independent responses from the model, and the message communicated is the empirical distribution over the answers. For both the verbalized and sampled probability exchange protocol, we measure the accuracy of the answer assigned the highest probability value. Full prompts are in Appendix E.3. All results use mean and 90% bootstrapped confidence over 100 mazes.

(R3) Calibration makes verbalized LLM probabilities useful. In our first experiment, we highlight the value of conversation calibration for verbalized probabilities. Recent work has shown that the probabilities verbalized by LLMs often fail to faithfully reflect the models' underlying uncertainty (Xiong et al., 2024; Kapoor et al., 2024). To illustrate the consequences of this miscalibration in a collaborative setting, we directly compare action-based and probability-based communication using prompted LLMs. We then apply post-hoc conversation calibration to the verbalized probabilities to test whether the lost information can be recovered through simple debiasing.

In Figure 6, verbalized probability exchange performs similarly to, or worse than, action exchange for GPT-5.4-nano, GPT-5.4-mini, and GPT-5, and yields only a small improvement for GPT-4.1-mini. Single verbal reports of uncertainty therefore do not appear to faithfully represent the model's uncertainty in a form that improves collaboration. Applying Algorithm 1 to verbalized probabilities consistently improves performance across all models. In particular, mean maze success rates at least double after four rounds of communication relative to the base verbalized probability protocol. These gains suggest that communicated uncertainty can support collaboration when the exchanged probabilities are first adjusted to account for model-specific miscalibration.

(R4) Sampled probabilities only partially elicit model uncertainty. Having established that verbalized probabilities are unreliable, we next compare against a stronger elicitation baseline: empirical probabilities from multiple sampled rollouts. This protocol estimates the model's action distribution directly from repeated responses, but is more expensive than a single verbalized probability report. We compare sampled against verbalized-probability exchange as well as post-hoc calibration.

(R5) Calibration explains when sampled probabilities help. GPT-4.1-mini behaves as expected: it performs poorly with verbalized probabilities but improves with sampled probabilities. For GPT-5.4-nano, however, sampled probabilities are on par with verbalized probabilities. This might be surprising at first glance, since we would expect the sampled probability measure to perform better according to our hypothesis. To understand why this might happen, we dig deeper into the distribution of both the verbalized and sampled probabilities in Figure **??**. We plot a histogram of maximum probability values under the verbalized and sampled probability exchange protocols, as well as the accuracy of the corresponding argmax action.

Here, we observe that GPT-4.1-mini's verbalized probabilities are miscalibrated, whereas its sampled probabilities

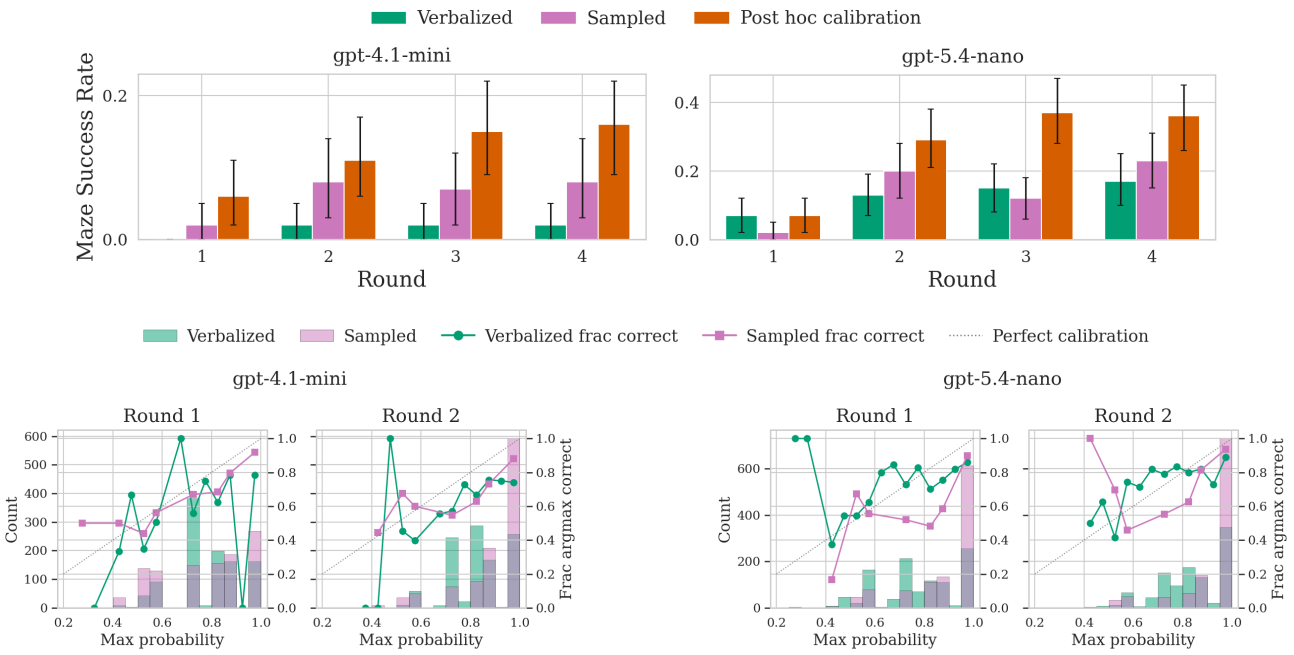

Figure 7. **Top row (R4):** Our post hoc calibration procedure matches or outperforms sampled probabilities. **Bottom row (R5):** Histogram of maximum assigned probabilities and corresponding argmax-action accuracy under verbalized and sampled probability exchange. For GPT-4.1-mini, sampled probabilities are better calibrated than verbalized probabilities and lead to stronger downstream performance. For GPT-5.4-nano, sampled probabilities are more overconfident, while verbalized probabilities are comparatively better calibrated.

are well calibrated. GPT-5.4-nano shows the opposite: its first-round sampled probabilities are often overconfident, while its verbalized probabilities are substantially better calibrated. This contrast further highlights the close relationship between calibration and downstream performance. Interestingly, the behavior of both models is consistent with broader trends in LLMs. As model capability improves, LLMs appear to become better at expressing uncertainty through verbalized probabilities (Xiong et al., 2024), which may help explain GPT-5.4-nano's improved verbal calibration. One possible explanation is that GPT-5.4-nano's sampled rollouts reflect stronger commitment due to more alignment training such as RLHF. This is consistent with prior work linking RLHF-style alignment to sharpened output distributions and increased overconfidence (Leng et al., 2025).

(R6) Communicating more information does not necessarily improve collaboration. Finally, we use GPT-4.1-mini to compare against an unconstrained communication baseline, in which each agent receives the previous agent's full response, including any generated reasoning. This is the setup studied by Davidson et al. (2025). We use the same prompt as in our other experiments, which asks the model for its uncertainty; therefore the communicated full response includes both the model's reasoning *and* verbalized probabilities. The results are found in Figure 8.

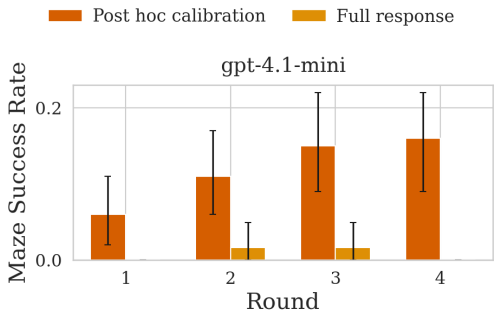

Figure 8. **(R6)** Communicating more information does not always help. Solely communicating probabilities produced by our post hoc calibration procedure leads to higher success than communicating the full response.

We see that unrestricted communication performs poorly as its success rate across all rounds is close to zero. This shows that simply providing the next agent with *more* information does not necessarily improve collaboration, and probability exchange can be more effective between agents than unconstrained natural-language communication, especially when long or noisy contexts make full-response communication difficult to use. We omit this baseline for other models, since we cannot access the generated reasoning.

Impact Statement

This paper studies how language-model agents can more reliably aggregate distributed information through calibrated probabilistic communication. As multi-agent LLM systems are increasingly deployed, improving their ability to communicate uncertainty may help make such systems more reliable. The purpose of this paper is to provide a proof of concept: our experiments are limited to stylized and maze-solving tasks, and we therefore do not study deployment risks directly.

References

Scott Aaronson. The complexity of agreement. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 634–643, 2005.

Robert J. Aumann. Agreeing to Disagree. *The Annals of Statistics*, 4(6):1236 – 1239, 1976. doi: 10.1214/aos/1176343654. URL https://doi.org/10.1214/aos/1176343654.

Robert J Aumann and Sergiu Hart. Agreeing on decisions. *Unpublished manuscript, The Einstein Institute of Mathematics, Jerusalem, Israel, http://math. huji. ac. il/˜ hart/-papers/agree. pdf (Last accessed: 17/04/2015)*, 2006.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 7066–7085, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.381. URL https://aclanthology.org/2024.acl-long.381/.

Natalie Collina, Surbhi Goel, Varun Gupta, and Aaron Roth. Tractable agreement protocols. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1532–1543, 2025.

Natalie Collina, Ira Globus-Harris, Surbhi Goel, Varun Gupta, Aaron Roth, and Mirah Shi. Collaborative prediction: Tractable information aggregation via agreement. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4712–4798. SIAM, 2026.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1538–1546. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/das19a.html.

Tim R Davidson, Adam Fourney, Saleema Amershi, Robert West, Eric Horvitz, and Ece Kamar. The collaboration gap. *arXiv preprint arXiv:2511.02687*, 2025.

A. P. Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. doi: 10.1080/01621459.1982.10477856.

Shrey Desai and Greg Durrett. Calibration of pre-trained transformers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 295–302, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.21. URL https://aclanthology.org/2020.emnlp-main.21/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11733–11763. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024. doi: 10.1038/s41586-024-07421-0. URL https://www.nature.com/articles/s41586-024-07421-0.

Jakob N. Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016. URL https://proceedings.

neurips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Dean P. Foster and Rakesh V. Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998. doi: 10.1093/biomet/85.2.379.

Rafael Frongillo, Eric Neyman, and Bo Waggoner. Agreement implies accuracy for substitutable signals. In *Proceedings of the 24th ACM Conference on Economics and Computation*, pages 702–733, 2023.

John D Geanakoplos and Heraklis M Polemarchakis. We can't disagree forever. *Journal of Economic theory*, 28 (1):192–200, 1982.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

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. In Kate Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8048–8057. International Joint Conferences on Artificial Intelligence Organization, August 2024. doi: 10.24963/ijcai.2024/890. URL https://doi.org/10.24963/ijcai.2024/890. Survey Track.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine M. Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Andrej Karpathy. NanoGPT. https://github.com/karpathy/nanoGPT, 2022.

Michael Kearns, Aaron Roth, and Emily Ryu. Networked information aggregation via machine learning. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4799–4845. SIAM, 2026.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VD-AYtP0dve.

Jixuan Leng, Chengsong Huang, Banghua Zhu, and Jiaxin Huang. Taming overconfidence in LLMs: Reward calibration in RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=l0tg0jzsdL.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Charles Michael Lewis, and Katia P. Sycara. Theory of mind for multi-agent collaboration via large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022. doi: 10.1162/tacl_a_00494. URL https://aclanthology.org/2022.tacl-1.50/.

Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434, 2023.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, et al. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*, 2023.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

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, pages 24824–24837, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Bryan Wilie, Samuel Cahyawijaya, Etsuko Ishii, Junxian He, and Pascale Fung. Belief revision: The adaptability of large language models reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

A. Additional Related Work

Calibration, scoring rules, and agreement-based aggregation. Our use of probabilistic messages is closely related to the classical literature on calibrated forecasting. Dawid (1982) formalized the idea that a forecaster's stated probabilities should match empirical frequencies, while Foster and Vohra (1998) showed that calibrated forecasting can be achieved constructively in repeated prediction. The theoretical backbone of our work also connects to agreement and information aggregation. Classical agreement results show that Bayesian agents with common priors cannot sustain common-knowledge disagreement (Aumann, 1976; Aumann and Hart, 2006), and that repeated exchange of beliefs can converge to agreement (Geanakoplos and Polemarchakis, 1982; Aaronson, 2005). Recent work replaces ideal Bayesian assumptions with tractable calibration-style conditions (Collina et al., 2025; 2026; Kearns et al., 2026). Frongillo et al. (Frongillo et al., 2023) show that agreement need not imply accuracy without additional signal-structure assumptions. Our contribution is complementary: we study how these ideas behave in LLM-agent collaboration, and empirically compare action messages, raw probabilistic messages, sampled probabilities, and post-hoc conversation-calibrated probabilities.

Uncertainty and calibration in language models. A large body of work shows that neural-network and language-model probabilities are not automatically calibrated. Guo et al. (2017) showed that modern neural networks can be substantially miscalibrated and that simple post-hoc temperature scaling can improve calibration. Desai and Durrett (2020) studied calibration in pretrained transformers such as BERT and RoBERTa, while Jiang et al. (2021) evaluated when generative language models' confidence estimates align with correctness. Zhao et al. (2021) showed that few-shot language-model predictions can be biased by prompt artifacts and proposed contextual calibration to mitigate these effects.

Recent work studies how to elicit uncertainty from large language models more directly. (Kadavath et al., 2022) evaluate whether language models can predict their own correctness, finding that self-evaluation can be informative but is format- and scale-dependent. (Lin et al., 2022) train models to express uncertainty in words, and (Mielke et al., 2022) study whether conversational agents' linguistic expressions of confidence are calibrated. (Tian et al., 2023) compare confidence-elicitation strategies for RLHF-trained models, while (Xiong et al., 2024) provide a broad empirical evaluation of verbalized, token-probability, and sampling-based confidence estimates. Kapoor et al. (2024) show that verbalized uncertainty in LLMs remains unreliable under prompting alone. These papers motivate our distinction between verbalized probabilities and probabilities that are

calibrated for use by another agent in a conversation. In addition, work on sycophancy (Perez et al., 2023; Sharma et al., 2023) motivates our biased-report model, where an agent's stated probability may be pulled toward a previous speaker's report rather than reflecting only its private evidence.

Sampling-based uncertainty and reasoning. Sampling multiple model outputs is another common way to estimate uncertainty. Chain-of-thought prompting (Wei et al., 2022) and self-consistency decoding (Wang et al., 2023) use multiple reasoning traces to improve single-agent reasoning by marginalizing over sampled answers. SelfCheckGPT (Manakul et al., 2023) uses consistency across sampled generations as a black-box signal for hallucination detection. (Kuhn et al., 2023) and (Farquhar et al., 2024) argue that uncertainty in language generation should often be measured over semantic equivalence classes rather than surface forms. Our sampled-probability protocol uses a related sampling signal, but for a different purpose: the empirical distribution over actions is communicated to another agent with complementary private information, rather than being used only to choose a single model's final answer.

LLM multi-agent collaboration. Our maze experiments build on the collaborative maze-solving benchmark of (Davidson et al., 2025), which demonstrates a "collaboration gap" between single-agent and multi-agent performance under partial observability. Other work studies multi-agent LLM interaction as a general reasoning or evaluation scaffold. (Du et al., 2024) use multi-agent debate to improve factuality and reasoning, (Chen et al., 2024) propose a round-table consensus procedure among diverse LLMs, and (Chan et al., 2024) use multi-agent debate for LLM-based text evaluation. (Guo et al., 2024) survey the broader literature on LLM-based multi-agent systems. Unlike these works, our agents have different private observations, and our main question is not whether discussion helps in aggregate, but which low-bandwidth message format preserves decision-relevant information.

Learned communication in multi-agent reinforcement learning. There is also a large literature on learning communication protocols in cooperative multi-agent reinforcement learning. (Foerster et al., 2016) introduce RIAL and DIAL, showing that agents can learn discrete or differentiable communication protocols in partially observable cooperative tasks. (Sukhbaatar et al., 2016) propose CommNet, where agents learn continuous communication alongside their policies. (Das et al., 2019) introduce TarMAC, which learns targeted communication by deciding both what to communicate and whom to address. Our work differs from this line by focusing on an interpretable communication channel—beliefs over actions—and by identifying conversation calibration as the condition under which such messages can be reliably used by another agent.

B. Warm-Up Example for Section 3

We include a two-bit example showing that best-action communication can discard information that is preserved by probability communication. Let the state $y = (y_1, y_2)$ be drawn from $\{00, 01, 10, 11\}$ with

$$\Pr(y = 00) = \frac{1}{4} - \epsilon, \qquad \Pr(y = 01) = \Pr(y = 10) = \Pr(y = 11) = \frac{1}{4} + \frac{\epsilon}{3}$$

where $0 < \epsilon < 1/4$. Alice observes y_1, Bob observes y_2, and the correct action $a^\star = f^*(y)$ is

y	00	01	10	11
$f^*(y)$	a_2	a_1	a_1	a_1.

Thus a_1 is correct except in the single state 00.

Suppose Alice speaks first and may send only her Bayes-optimal action. If $y_1 = 1$, she knows the state is either 10 or 11, so a_1 is certainly correct: $\Pr(a^\star = a_1 \mid y_1 = 1) = 1$. If $y_1 = 0$, Alice compares the two states 00 and 01, and still strictly prefers a_1:

$$\Pr(a^\star = a_1 \mid y_1 = 0) = \Pr(y = 01 \mid y_1 = 0) = (\frac{1}{4} + \frac{\epsilon}{3})/(\frac{1}{2} - \frac{2\epsilon}{3}) > \frac{1}{2}.$$

Therefore, she sends the same action, a_1, no matter which bit she observes: The action message carries no information about y_1. Bob faces the same situation symmetrically, so an action-only conversation cannot identify the exceptional state 00; the agents play a_1, which is correct with prior probability $3/4 + \epsilon$.

Under probability communication, Alice instead sends her posterior probability that a_1 is correct. This message equals 1 when $y_1 = 1$ and equals $(\frac{1}{4} + \frac{\epsilon}{3})/(\frac{1}{2} - \frac{2\epsilon}{3})$ when $y_1 = 0$. These two values are distinct, so the probability message reveals her bit. Since Bob already observes y_2, he can identify the full state and report the correct action with probability one.

The lesson is that the separation is not between actions and probabilities as syntactic objects, but between messages that preserve distinct posterior states and messages that collapse all evidence states inducing the same best action.

C. Proofs for Section 3

We start with a recurring observation used throughout. For shorthand, let

$$u_k := \sum_{j \le k} \alpha_j S_j, \qquad r_k := \sum_{j > k} \alpha_j S_j,$$

with $u_0 := 0$ and $r_0 := p^\star - 1/2$, so that $p^\star = 1/2 + u_k + r_k$ for every k.

Lemma C.1 (Conditional vanishing)**.** *Let $\mathcal{G}$ be a σ-algebra and suppose b_i is independent of $\sigma(\mathcal{G}, \{b_\ell : \ell \neq i\})$ for some $i \in [K]$. For any random variable W that is a function of $\{b_\ell : \ell \neq i\}$, $\mathbb{E}[b_i W \mid \mathcal{G}] = \mathbb{E}[b_i]\,\mathbb{E}[W \mid \mathcal{G}] = 0$. In particular, if $J \subseteq [K]$ is nonempty and some $i \in J$ is independent of $\sigma(\mathcal{G}, \{b_\ell : \ell \neq i\})$, then $\mathbb{E}[\prod_{\ell \in J} b_\ell \mid \mathcal{G}] = 0$.*

Proof. Since W is measurable with respect to $\sigma(\{b_\ell : \ell \neq i\})$, it is also measurable with respect to $\sigma(\mathcal{G}, \{b_\ell : \ell \neq i\})$. By the assumed independence,

$$\mathbb{E}[b_i \mid \sigma(\mathcal{G}, \{b_\ell : \ell \neq i\})] = \mathbb{E}[b_i] = 0.$$

Therefore

$$\mathbb{E}[b_i W \mid \mathcal{G}] = \mathbb{E}[W\,\mathbb{E}[b_i \mid \sigma(\mathcal{G}, \{b_\ell : \ell \neq i\})] \mid \mathcal{G}] = 0.$$

For the product claim, write $\prod_{\ell \in J} b_\ell = b_i W$ with $W = \prod_{\ell \in J \setminus \{i\}} b_\ell$. □

C.1. Proof of Theorem 3.2

Proof. We prove $p_k = 1/2 + u_k$ by induction on k. For $k = 1$, the previous message is empty; Alice knows b_1, and every later term S_j with $j \geq 2$ contains Bob's unseen bit b_2, so Lemma C.1 gives $\mathbb{E}[S_j \mid x_A] = 0$ and therefore $p_1 = 1/2 + \alpha_1 b_1 = 1/2 + u_1$.

Inductively assume $p_{k-1} = 1/2 + u_{k-1}$. By Assumption 3.1, the round-k speaker recovers the prefix $(S_1, \ldots, S_{k-1})$ from the scalar p_{k-1}; combined with its own bit b_k, the speaker also determines $S_k = S_{k-1} b_k$. Every later S_j with $j \geq k+1$ contains the next bit b_{k+1}, which has opposite parity from the speaker, so Lemma C.1 forces its conditional mean to vanish. Hence

$$p_k = \mathbb{E}[p^\star \mid x_{\sigma(k)}, p_{k-1}] = \frac{1}{2} + \textstyle\sum_{j \leq k} \alpha_j S_j = \frac{1}{2} + u_k.$$

For the risk identity, $a^\star$ is a $\mathrm{Bernoulli}(p^\star)$ draw given b, so for any predictor q measurable with respect to the public summary and the speaker's view, $\mathbb{E}[a^\star \mid b, q] = p^\star$ and the cross term $\mathbb{E}[(a^\star - p^\star)(p^\star - q)] = 0$. Squaring $a^\star - q = (a^\star - p^\star) + (p^\star - q)$ and subtracting the $q = p^\star$ case gives

$$\mathbb{E}[(a^\star - q)^2] - \mathbb{E}[(a^\star - p^\star)^2] = \mathbb{E}[(p^\star - q)^2].$$

Taking $q = p_k$, we have $p^\star - p_k = r_k = \sum_{j > k} \alpha_j S_j$; Lemma C.1 applied to each cross term $\mathbb{E}[S_j S_{j'}]$ with $j \neq j'$ yields $\mathbb{E}[r_k^2] = \sum_{j > k} \alpha_j^2$.

For the one-step improvement, let $\mathcal{G}_k := \sigma(x_{\sigma(k)}, p_{k-1})$. Since p_{k-1} is $\mathcal{G}_k$-measurable and $p_k = \mathbb{E}[a^\star \mid \mathcal{G}_k]$, the L^2 projection identity gives $\mathbb{E}[(a^\star - p_{k-1})^2] = \mathbb{E}[(a^\star - p_k)^2] + \mathbb{E}[(p_k - p_{k-1})^2]$, and $p_k - p_{k-1} = \alpha_k S_k$ has squared expectation α_k^2. □

Remark C.2 (Full history vs. previous-message sufficiency). The distinct-partial-sums condition in Assumption 3.1 is used only to make the previous scalar message a sufficient summary of the public history. Dropping it gives a weaker protocol whose rounds still reduce risk by $\mathbb{E}[(p_k - p_{k-1})^2]$, but without a closed-form per-round improvement. Alternatively, sending the full message history in place of just p_{k-1} restores the closed-form improvement α_k^2, because the increment $p_k - p_{k-1} = \alpha_k S_k$ always reveals the next parity.

Remark C.3 (Loss-adaptive stopping vs. exact recovery of $h^\star$). To achieve squared-loss excess at most ε it suffices to run until $\sum_{j > k} \alpha_j^2 \leq \varepsilon$, which can be much smaller than K. Exact recovery of $h^\star$ may instead require all K rounds: if $\sum_{j \leq k} \alpha_j < \sum_{j > k} \alpha_j$ for every $k < K$, the unrevealed tail can flip the sign of $u_k + r_k$, so no $k < K$ prediction message determines $h^\star$.

C.2. Proof of Theorem 3.3

Proof. For the parity algebra, let $\tilde{p}_k$ be the round-k speaker's posterior under the action-channel transcript, and let $A_k := 2\,\mathbf{1}\{\tilde{p}_k \geq 1/2\} - 1 = (\tilde{p}_k - 1/2) \in \{-1, +1\}$ be the ± 1 recoding of the round-k action message. The claim in Theorem 3.3 that every message equals $\mathbf{1}\{b_1 = +1\}$ is equivalent to $A_k = b_1$ for every k.

We show $A_k = b_1$ by induction. At round 1, Alice has no previous message and knows b_1; every S_j with $j \geq 2$ contains Bob's unseen b_2 and vanishes in conditional expectation by Lemma C.1, so her posterior is $1/2 + \alpha_1 b_1$ and $A_1 = (\alpha_1 b_1) = b_1$.

Suppose all previous action messages equal b_1, so the public transcript reveals no information beyond the first bit. If Alice speaks at round k, she still lacks b_2, so the same conditional-expectation calculation gives $\tilde{p}_k = 1/2 + \alpha_1 b_1$ and $A_k = b_1$. If Bob speaks, he knows b_2; every S_j with $j \geq 3$ contains the unseen bit b_3 and averages out, so his posterior is

$$\tilde{p}_k = \frac{1}{2} + \alpha_1 b_1 + \alpha_2 b_1 b_2 = \frac{1}{2} + b_1\big(\alpha_1 + \alpha_2 b_2\big).$$

Since $\alpha_1 > \alpha_2$, the factor $\alpha_1 + \alpha_2 b_2$ is positive for both values of b_2, so $A_k = b_1$. Hence $A_k = b_1$ at every round and the induced decision does not change after round 1.

For the constant-gap instance, take $K = 3$, $\alpha_1 = 0.20$, $\alpha_2 = 0.15$, $\alpha_3 = 0.10$, which satisfy $\alpha_1 > \alpha_2$ and $\alpha_1 + \alpha_2 + \alpha_3 < 1/2$. Theorem 3.2 gives $p_3 = p^\star$. Meanwhile, the action message at every round equals $\mathbf{1}\{b_1 = +1\}$ (equivalently, $A_k = b_1$). On $E := \{b_2 = -1, b_3 = +1\}$, which has probability $1/4$, the prefix parities are $S_1 = b_1$ and $S_2 = S_3 = -b_1$, so $p^\star - 1/2 = b_1(\alpha_1 - \alpha_2 - \alpha_3) < 0$; hence $h^\star = \mathbf{1}\{p^\star \geq 1/2\} = \mathbf{1}\{b_1 = -1\}$, which disagrees with the action message on E. Thus $\Pr[\hat{a}^k \neq h^\star] \geq 1/4$ at every round. □

D. Proofs for Section 4

Throughout this appendix section, we work in the chained-parity model of Section 3 under Assumption 3.1. We begin with the following observation:

Observation D.1 (Learning cross-calibrated forecasts)**.** *Let Z be any information available at prediction time, such as a pair of communicated forecasts $Z = (\hat{p}_A^k, \hat{p}_B^\ell)$. Suppose the predictor class can represent the true conditional label probabilities. If the training loss is one whose best population prediction is the true conditional probability, such as cross-entropy or squared loss for binary probabilities, then training a predictor h to predict $a^\star$ from Z gives*

$$h^\star(Z)(a) = \Pr(a^\star = a \mid Z).$$

Therefore the learned forecast $\tilde{p} = h^\star(Z)$ is calibrated even after conditioning on Z: $\Pr(a^\star = a \mid \tilde{p}, Z) = \tilde{p}(a)$ almost surely. Moreover, if $Z = (W, \hat{q})$ includes a previous communicated forecast $\hat{q}$, then also $\Pr(a^\star = a \mid \tilde{p}, \hat{q}) = \tilde{p}(a)$ almost surely. Thus loss minimization on features that include the previous forecast learns a forecast that is cross-calibrated with respect to that forecast.

Proof. By the assumption on the loss, the best population predictor outputs the true conditional label probabilities, so $h^\star(Z)(a) = \Pr(a^\star = a \mid Z)$. Let $\tilde{p} = h^\star(Z)$. Since $\tilde{p}$ is a function of Z,

$$\Pr(a^\star = a \mid \tilde{p}, Z) = \Pr(a^\star = a \mid Z) = \tilde{p}(a).$$

If $Z = (W, \hat{q})$, then averaging the last display over the remaining information W, while keeping $(\tilde{p}, \hat{q})$ fixed, gives

$$\Pr(a^\star = a \mid \tilde{p}, \hat{q}) = \mathbb{E}[\Pr(a^\star = a \mid Z) \mid \tilde{p}, \hat{q}] = \mathbb{E}[\tilde{p}(a) \mid \tilde{p}, \hat{q}] = \tilde{p}(a).$$

□

We then expand the calibration claim of Theorem 4.1 into self- and cross-calibration (Definitions 2.1 and 2.2), and quantify the excess squared loss from using raw reports $\hat{p}_k$ in place of p_k.

We first consider the direct-use benchmark in which the receiver treats the raw report $\hat{p}_k$ itself as the round-k prediction. This benchmark isolates the squared-loss cost of reporting a biased scalar before applying any post-hoc calibration or projection rule.

Proposition D.2 (Self-calibration removes global miscalibration)**.** *If $\lambda = 0$ in (1), then self-calibration already recovers the ideal scalar: $\Pr(a^\star = 1 \mid \hat{p}_k) = p_k$. Using raw $\hat{p}_k$ in place of p_k incurs excess squared loss*

$$\mathbb{E}[(\hat{p}_k - p_k)^2] = (c-1)^2\, \mathbb{E}\left[\left(p_k - \tfrac{1}{2}\right)^2\right] = (c-1)^2 \sum_{j \le k} \alpha_j^2.$$

Proof. When $\lambda = 0$, $\hat{p}_k = 1/2 + c(p_k - 1/2)$ is invertible: $p_k = 1/2 + (\hat{p}_k - 1/2)/c$. Since $p^\star = p_k + r_k$ and $\hat{p}_k$ is a function of $b_1, \ldots, b_k$, Lemma C.1 gives $\mathbb{E}[r_k \mid \hat{p}_k] = 0$, so $\Pr(a^\star = 1 \mid \hat{p}_k) = \mathbb{E}[p^\star \mid \hat{p}_k] = p_k$. The raw-report loss is $\mathbb{E}[(\hat{p}_k - p_k)^2] = (c-1)^2 \mathbb{E}[(p_k - 1/2)^2] = (c-1)^2 \sum_{j \le k} \alpha_j^2$. □

Proposition D.3 (Cross-calibration removes previous-message bias)**.** *For any $c > 0$ and $\lambda \in \mathbb{R}$ in (1), the conditional label distribution given the raw report and the previous agent's report recovers the ideal scalar:*

$$\Pr(a^\star = 1 \mid \hat{p}_k, p_{k-1}) = p_k.$$

Thus a post-hoc forecast learned from $(\hat{p}_k, p_{k-1})$ is cross-calibrated with respect to p_{k-1} and outputs p_k. Using raw $\hat{p}_k$ in place of p_k incurs excess squared loss

$$\mathbb{E}[(\hat{p}_k - p_k)^2] = (c-1)^2 \sum_{j \le k} \alpha_j^2 + \big(2(c-1)\lambda + \lambda^2\big) \sum_{j < k} \alpha_j^2.$$

Proof. For any λ, observing $(\hat{p}_k, p_{k-1})$ determines

$$p_k = \tfrac{1}{2} + \tfrac{1}{c}\left[\left(\hat{p}_k - \tfrac{1}{2}\right) - \lambda\left(p_{k-1} - \tfrac{1}{2}\right)\right],$$

so $p^\star = p_k + r_k$ with $\mathbb{E}[r_k \mid \hat{p}_k, p_{k-1}] = 0$ by Lemma C.1. Hence $\Pr(a^\star = 1 \mid \hat{p}_k, p_{k-1}) = \mathbb{E}[p^\star \mid \hat{p}_k, p_{k-1}] = p_k$. If the raw report is used directly then $\hat{p}_k - p_k = (c-1)(p_k - 1/2) + \lambda(p_{k-1} - 1/2)$. With $p_k - 1/2 = u_k$ and $p_{k-1} - 1/2 = u_{k-1} = u_k - \alpha_k S_k$, cross-term cancellation via Lemma C.1 gives $\mathbb{E}[u_k^2] = \sum_{j \le k} \alpha_j^2$ and $\mathbb{E}[u_{k-1}^2] = \mathbb{E}[u_k u_{k-1}] = \sum_{j < k} \alpha_j^2$, yielding the stated identity.

□

Proposition D.3 is exactly the chained-parity instance of Observation D.1: learning $\Pr(a^\star = 1 \mid \hat{p}_k, p_{k-1})$ from labelled transcripts produces a cross-calibrated forecast that equals p_k, and hence recovers Theorem 3.2 regardless of the raw report's confidence distortion (c) or anchoring to the previous message (λ). This proves Theorem 4.1.

Round-by-round behavior under projection. Propositions D.2 and D.3 say that, given enough labelled data, calibration recovers the ideal scalar p_k from a biased report. We now analyze a different receiver model: a receiver that does not know the bias model and therefore treats $\hat{p}_k$ as if it were the true posterior. When $\hat{p}_k$ is not a possible message under any valid speaker posterior, the receiver rounds $\hat{p}_k$ to the closest valid message before updating. Formally, if S is the set of valid speaker messages, the receiver's Bayesian update is computed with respect to

$$\bar{p}_k \;\in\; \arg\min_{s \in S} |\hat{p}_k - s|. \tag{2}$$

The set S should be read as the *receiver-consistent* valid set: the posterior values that could have been sent by the speaker

among worlds consistent with the receiver's own private observation and the public transcript so far, which avoids assigning positive probability to a message the receiver already knows to be impossible.

Under this rule, bias is harmless until it pushes $\hat{p}_k$ across the midpoint between two adjacent valid messages, at which point the receiver decodes the wrong latent bit and pays a discrete cost. We work with the original biased-report model of (1), with the same (c, λ) across rounds, and trace what happens round by round.

Round 1. Alice observes b_1 and her ideal posterior is $p_1 = \frac{1}{2} + \alpha_1 b_1$. From Bob's view the valid messages are $\{\frac{1}{2} - \alpha_1, \frac{1}{2} + \alpha_1\}$, symmetric around $p_0 = \frac{1}{2}$ at distance α_1. Since $p_0 = \frac{1}{2}$, the anchoring term in (1) vanishes and $\hat{p}_1 - \frac{1}{2} = c\alpha_1 b_1$ has the same sign as b_1 for every $c > 0$. Nearest-valid projection therefore recovers b_1, and the round-1 message conveys the same information it would under a calibrated speaker.

Round 2. Conditional on correct round-1 decoding, Bob now sees $p_1 = \frac{1}{2} + \alpha_1 b_1$ as a public scalar and observes his own bit b_2. His ideal posterior is $p_2 = p_1 + \alpha_2 b_1 b_2$, and from Alice's view the valid round-2 messages are $\{p_1 - \alpha_2, p_1 + \alpha_2\}$, symmetric around p_1 at distance α_2. Nearest-valid projection of $\hat{p}_2$ recovers p_2 iff $\hat{p}_2$ lies on the same side of p_1 as p_2 (with equality corresponding to a tie). Unlike at round 1, the biased report now picks up an anchoring contribution from $p_{k-1} = p_1$, and this is the source of failure.

Proposition D.4 (Decoding failure at round 2)**.** *Under the biased reports of* (1) *with* $c > 0$*, conditional on correct round-*1 *decoding, nearest-valid projection at round* 2 *decodes incorrectly on the event* $\{b_2 = -(c + \lambda - 1)\}$*, of probability* $1/2$*, whenever*

$$|c + \lambda - 1|\, \alpha_1 \;>\; c\, \alpha_2.$$

A wrong decode flips $\alpha_2 S_2$ *in the receiver's projected posterior. Conditional on the wrong-decode event, this contributes squared-loss excess* $4\alpha_2^2$ *relative to the ideal round-*2 *posterior; averaged over* b_2 *conditional on correct round-*1 *decoding, the contribution is* $2\alpha_2^2$.

Proof. Conditional on correct round-1 decoding, $p_1 = \frac{1}{2} + \alpha_1 b_1$ and $p_2 = p_1 + \alpha_2 b_1 b_2$, so (1) gives

$$\hat{p}_2 - p_1 = c\alpha_2 b_1 b_2 + (c + \lambda - 1)\alpha_1 b_1.$$

Correct decoding requires this to have the same sign as $b_1 b_2$; multiplying by $b_1 b_2$ gives the equivalent condition $c\alpha_2 + (c + \lambda - 1)\alpha_1 b_2 > 0$, which fails on the unfavorable sign $b_2 = -(c + \lambda - 1)$ exactly when $|c + \lambda - 1|\alpha_1 > c\alpha_2$ (with equality corresponding to a tie in nearest-valid projection). On the wrong-decode event, projection flips $\alpha_2 S_2$ to $-\alpha_2 S_2$, so the projected posterior differs from the ideal round-2 posterior by $2\alpha_2 S_2$. By the squared-loss identity used in Theorem 3.2, the conditional excess is $(2\alpha_2)^2 = 4\alpha_2^2$. Since the wrong-decode event has probability $1/2$ conditional on correct round-1 decoding, its expected contribution is $2\alpha_2^2$. □

This parallels Theorem 3.3: action communication transmits b_1 correctly and then stalls; affine-biased probability communication also transmits b_1 correctly under projection, but begins to lose b_2 once the imbalance condition $|c + \lambda - 1|\alpha_1 > c\alpha_2$ holds. The round-2 projection failure disappears when $c + \lambda = 1$, while true calibration of the affine report corresponds to the stronger condition $c = 1$ and $\lambda = 0$. The two parts of the appendix are thus complementary: Propositions D.2–D.3 say that post-hoc calibration suffices to recover the ideal scalar p_k, while Proposition D.4 identifies a regime in which an uncalibrated projection-based receiver flips the second latent bit.

E. Experimental details

Here, we provide additional information on reproducibility of our experiments.

E.1. Task design

Chained Parities For the parity task, we construct our dataset as prescribed by the theory in Section 3. We choose a total of 30 coefficients. We set the first four values to $[0.2, 0.15, 0.1, 0.02]$ and assign the remaining 26 values equally so that $\sum_j \alpha_j = 0.49$. The bits of each datapoint are split disjointly between the two agents in alternating order, and the binary label is sampled from $P(Y{=}1 \mid b) = \frac{1}{2} + \sum_j \alpha_j s_j$, where s_j is the j-th prefix product of the hidden signs. Then, we generate a total of two million datapoints which we split into a 90% train and 10% test split. All results are reported on the test split.

Mazes For the maze task, we construct mazes similar to those introduced by (Davidson et al., 2025). We consider mazes of size 6×6. For evaluation purposes, we only consider mazes with a single correct path. The wall density is 30%. Each agent sees exactly half of the maze cells, assigned uniformly at random, and the agents receive disjoint masks so that each maze is always solvable from the joint information. An example maze is shown below. Again, we generate two million datapoints in total, split the data into a 90% train and 10% test split, and report our results on the test set. The data contains one example per step along the unique solution: the agent's masked grid is rendered with the current position marked, and the correct next move is used as the label. There are four answer tokens that can represent the true label: up, down, left, and right.

```
@##.#.
.##...
...#..
.#.##.
#..###
.#...*
```

(a) x

```
@#??#?
?#??.?
?.??.?
.?.##?
#..##?
.#.??*
```

(b) x_A

```
@?#.?.
.?#.?.
.?.#?.
?#???.
?????#
???..*
```

(c) x_B

Figure 9. Example of full maze and agents' private observations.

E.2. Transformer experiments

For both tasks we use a small, character-level GPT based on the NanoGPT implementation (Karpathy, 2022): 4 transformer blocks, 6 attention heads, embedding dimension 240, and a causal mask. Each agent maintains its own per-task vocabulary derived from the raw character set of its inputs, so all sequences are tokenized at the character level. The context length is set to `prefix_size` $+ 1$, where the prefix is the serialized example up to (but not including) the answer; any annotations appended by a collaborator (see below) extend the prefix accordingly. We treat each task as next-token prediction with the loss measured only at the answer position(s).

Per-agent inputs. For chained parity, agent 0 ("Bob") sees the odd-indexed bits and agent 1 ("Alice") sees the even-indexed bits; the unobserved positions are replaced with a mask token, and the label bit is appended after an = delimiter. For mazes, each agent's input is the flattened 6×6 grid restricted to its assigned half of the cells, with the current position marked by `@`, followed by = and the next move from $\{d, r, u, l\}$. A single maze with path length ℓ contributes ℓ training examples, each predicting the next move from the current position.

Conversation loop. We instantiate our protocol as iterative training across rounds: in round r, agent $r \bmod K$ (with K the number of agents) trains a fresh transformer from scratch on its currently annotated dataset. After a fixed number of optimizer steps, the trained agent runs over its peer's round-0 inputs and emits a message for each example, which becomes part of the prefix that the next agent will train on in round $r{+}1$. Because each round trains a freshly initialized model, any improvement in joint performance must come through the appended messages rather than transferred weights.

Optimization. Each agent is trained from scratch with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.99$, weight decay 0.1, gradient clip 1.0) in bfloat16 mixed precision on a single GPU, with batch size 1024 and no gradient accumulation. The learning rate uses linear warmup over 100 steps to a peak of 10^{-4}, followed by cosine decay to 10^{-5} over the remainder of training. We run 2000 iterations per round on chained parity (5 rounds) and 50,000 iterations per round on mazes (4 rounds); in both cases we keep the checkpoint with the best validation loss within each round.

E.3. LLM experiments

The LLM experiments follow the same per-round procedure as our transformer experiments, with each agent now replaced by a pretrained LLM queried through the API rather than a model trained from scratch. Unless otherwise noted we use temperature 1.0, run 4 rounds of conversation per maze, and evaluate on $N = 100$ mazes drawn from the held-out test split. At each round, the active agent receives a prompt containing (i) a legend-annotated rendering of its own masked 6×6 maze, (ii) the sequence of moves taken so far (from which the current position can be inferred), and (iii) the other agent's most recent response from the previous round. We consider three communication settings, which differ only in what part of the partner's response is appended to the next prompt: the full free-form message, a single move token in $\{d, r, u, l\}$, or a probability vector over the four moves. Probabilities can either be elicited directly from the model in the response, post-calibrated from the verbalized answer, or obtained from rollouts. The agent is asked to produce a chain-of-thought trace followed by a single move after the delimiter `~`, and – in the verbalized setting – a probability vector after the delimiter `%`. The joint trajectory advances by one step per round; a single incorrect move terminates the maze, and we report the resulting per-maze success rate.

The prompt we use to query the LLM at each step is:

```
Task: You are going to play the
    collaborative maze game together with
    another agent. The maze consists of a
    grid with walls and a goal. Your task
    is to jointly determine the next move
    on the path. You and the other agent
```

```
    will take that action together. You
    will each get your own map of the same
    maze, with some coordinates hidden.
    Because of the hidden coordinates, you
    will need to communicate with the other
     agent to share information about the
    maze and coordinate your next move.
    Both agents together have enough
    information to solve the maze, so you
    do not need to explore.
Rules:
- You can only move to adjacent cells [down
    (d), right(r), up(u), left(l)].
- You can not move diagonally or through
    walls.
- Once you make an incorrect move, you lose.

Here is **your** map of the maze with a
    legend of the symbols:
@?#.?.
.?#.?.
.?.#?.
?#???.
?????#
???..*
Legend:
@ - Current Position
* - Goal Position
. - Path
# - Wall
? - Hidden Cell

You must answer with one of d, r, u, l. Put
     your final answer after the delimiter
    ~. For example, if your final answer is
     d, your response should contain ~d.

Finally, give the probabilities you think
    each move d, r, u, l is correct after
    the delimiter %. For example, if you
    think the probability of d, r, u, l is
    0.8, 0.1, 0.05, 0.05, your response
    should contain %[0.8, 0.1, 0.05, 0.05].
     Make sure it is a valid probability
    vector, i.e. the sum of the
    probabilities is 1.

Your moves are timed for speed so hurry up!
```