DO LLMS ACT ON WHAT THEY KNOW? FROM PARTNER REPRESENTATIONS TO COOPERATIVE ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperation with unfamiliar partners requires adapting to communication conventions that are not known in advance. We study this problem in a controlled Hanabi-derived environment with scripted hint generation, LLM-controlled receiving decisions, and frozen model weights. Across eight LLMs, linear probes recover intent conventions substantially more accurately than target conventions, yet receiving choices do not consistently agree with the sender's convention. We compare probe-predicted and ground-truth conventions presented either as general rules or as externally computed action recommendations. Rule statements yield modest and model-dependent changes in cooperation, whereas action translation produces larger gains on average. In a Qwen3-8B case study, matched-state statement reversals reveal much greater sensitivity to action recommendations than to rule statements. Activation transfers from oracle-action and non-oracle hint-restatement donors improve intent accuracy on both action classes, but the tested alternatives do not reliably reproduce these benefits. Together, these results distinguish convention decodability, sensitivity to convention information, and cooperative performance, and highlight limitations in turning available partner information into receiving decisions.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as decision makers that carry out tasks with humans or other agents. In these settings, success depends not only on solving a task individually but also on coordinating choices with a partner. A model must account for what its partner intends, how the partner acts, and how its own actions will be interpreted. Even LLMs with strong task knowledge and reasoning abilities can struggle with these demands (Agashe et al., 2025; Ramesh et al., 2026; Liang et al., 2025; Yao et al., 2025). Understanding LLM cooperation therefore requires examining how information about a partner informs the model's decisions.

This requirement becomes particularly important when cooperating with an unfamiliar partner. Partners may share a goal while using different strategies or communication conventions, making a previously effective response inappropriate in a new interaction (Bard et al., 2020; Carroll et al., 2019). An agent must use observed behavior to infer relevant partner characteristics and adjust its subsequent choices. Zero-shot coordination and ad hoc teamwork provide important settings for studying cooperation with partners that were not encountered during joint training (Hu et al., 2020; Dizdarević et al., 2025). Within this broader problem, we focus on how LLMs use interaction history to adapt their behavior without partner-specific updates to the base model's weights.

Prior work incorporates explicit reasoning about partners into LLM planning (Li et al., 2023a; Zhang et al., 2024b). ProAgent estimates teammate intentions and updates its beliefs from observed behavior (Zhang et al., 2024a), while Hypothetical Minds generates, evaluates, and revises natural-language hypotheses about partner strategies (Cross et al., 2025). However, accurate partner prediction need not translate into behavioral adaptation (Riemer et al., 2025). In communication-based cooperation, this distinction matters because interpreting a signal may require applying an estimate of the partner's convention. For LLM agents, behavioral adaptation therefore depends not only on forming such an estimate, but also on whether that information influences the final decision. We ask whether partner conventions are decodable from the model's hidden representations and whether making that information available changes how the model interprets the current hint.

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

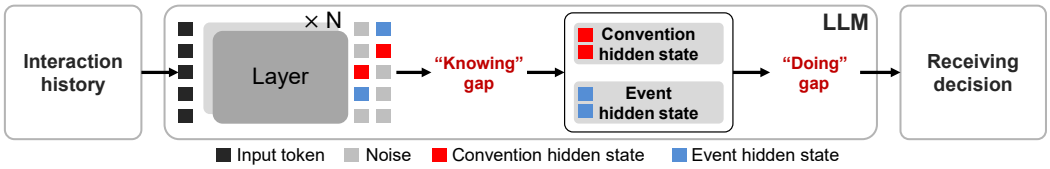


Figure 1: **Conceptual overview from interaction history to receiving decision.** *Knowing* measures how accurately the sender's convention can be decoded from the receiver's hidden states, while *Doing* measures whether the receiver's target and intent choices agree with the sender's convention. The internal components are schematic rather than identified model subspaces.

To study these questions, we use a controlled environment derived from Hanabi in which agents have different communication conventions (Bard et al., 2020). The same hint can refer to different cards or request different decisions depending on the sender's convention. Each agent retains its own sending convention while receiving hints from a partner whose convention is not explicitly disclosed. We collect probing data from scripted interactions and evaluate interventions in games where language models make receiving decisions while hint generation remains scripted. This design focuses the analysis on convention-dependent interpretation and action selection while controlling the sending policy.

To distinguish convention inference from its application, we probe the receiver's hidden representations for the partner's assigned convention (Hewitt & Liang, 2019; Belinkov, 2022). We then evaluate textual and activation-based interventions to test whether decodable convention information can influence action selection, while keeping the base LLM frozen throughout (Zhang & Nanda, 2024; Rimsky et al., 2024). We separately evaluate *knowing*, whether the sender's convention can be decoded from the receiver's representations, and *doing*, whether the receiver's choices follow that convention. We report game score separately as the resulting cooperative outcome.

Our experiments show that partner-convention information is more readily decoded on the intent axis than on the target axis, yet receiving decisions do not consistently follow the partner's convention. Providing convention information as a rule has limited effects on cooperation, whereas translating it into the current decision yields larger gains. Matched-state analyses show much stronger responses to action recommendations than to rule statements. Activation-based analyses further show that some state-specific donor transfers improve both action classes, whereas simpler transformations do not preserve these gains. Our contributions are threefold:

- We develop a controlled cooperation setting that separates sender-specific convention inference, interpretation of the current hint, and receiving decisions.
- We show that convention information presented as a rule has limited effects on cooperation, whereas translating it into action recommendations yields larger gains.
- We use matched-state and activation-based analyses to distinguish decodable partner information from its behavioral use and from changes in default action preference.

2 Related Work

Partner-aware coordination. Successful cooperation requires agents to coordinate with partners whose strategies, conventions, or preferences may differ from their own. Prior work has studied robust coordination through symmetry-based training (Hu et al., 2020), cross-environment and population objectives (Jha et al., 2025; Hui et al., 2026), and general formulations for unseen teammates (Dizdarević et al., 2025). A related line grounds cooperation in human behavior, from imitation-based methods (Carroll et al., 2019; Jacob et al., 2022; Meta Fundamental AI Research Diplomacy Team (FAIR) et al., 2022) to influence-based team steering (Sheng & Paleja, 2026). More recently, LLM agents have been evaluated in cooperative games (Agashe et al., 2025; Liang et al., 2025; Ramesh et al., 2026) and equipped with explicit partner models (Zhang et al., 2024a; Cross et al., 2025; Li et al., 2023a; Zhang et al., 2024b; Goel et al., 2026). Related work studies in-context co-player inference in sequence-model agents (Weis et al., 2026) and belief tracking and partner generalization in multi-agent environments (Ruhdorfer et al., 2026). Other work questions what behavioral evaluations alone establish (Riemer et al., 2025; Chandak et al., 2025; Singh et al.,

2025). Behavioral performance alone does not reveal whether partner information is absent or simply unused. Our interventions instead ask whether feeding the readout back changes decisions in line with the partner's convention rather than merely shifting default action tendencies.

Probing internal representations. Probing studies have recovered latent world-state information from game-playing models (Li et al., 2023b; Nanda et al., 2023; Karvonen, 2024; Kamel et al., 2025) and cooperative policies (Tessera et al., 2026), while belief representations have been shown to support intervention (Sarfati et al., 2026). Related work also identifies internal partner models (Mon-Williams et al., 2025), belief-tracking mechanisms (Prakash et al., 2026), and discrepancies between internal or intermediate information and final behavior (Lekeas & Stamatopoulos, 2026; Young, 2026; Boppana et al., 2026; Li et al., 2026b; Arghal et al., 2026). Because probing can suffer from confounds, unselective readouts, and statistical instability (Hewitt & Liang, 2019; Belinkov, 2022; Nordby et al., 2026; Méloux et al., 2025), we pair each readout with a behavioral intervention on the same decisions, comparing whether partner information is readable and whether it influences behavior.

Activation patching and steering. Activation patching tests causal structure by transplanting internal states between runs (Zhang & Nanda, 2024), with recent work extending it to circuit-level analysis (Haklay et al., 2025; Wu et al., 2026). Steering instead adds learned or synthesized directions at inference time (Rimsky et al., 2024), including applications to game strategies and theory-of-mind behavior (Sun & Zhang, 2026; Li et al., 2026a), though its robustness remains an active concern (Tan et al., 2024; Im & Li, 2025; Braun et al., 2026; Nguyen et al., 2026).

3 Task and Experimental Setup

3.1 Convention-Based Cooperation Task

We study how LLMs use interaction history to interpret a partner's communication and adapt their choices. Our environment is derived from Hanabi, a cooperative card game in which each player observes the partner's hand but not their own (Bard et al., 2020). Players can play a card, discard a card, or spend a hint token to identify all cards of a particular color or rank in the partner's hand. The game rules determine which cards a hint marks, while a communication convention specifies the intended response. We use three colors with five ranks per color, so completing all color stacks yields a maximum shared score of 15. Playing a card that is not the next rank in its color stack costs one shared life, and the episode terminates when no lives remain. The environment is implemented using the Hanabi Learning Environment, with game parameters listed in Appendix B.

Each player $P \in \{A, B\}$ receives a sending convention $\theta_P = (\theta_P^{\text{target}}, \theta_P^{\text{intent}})$ that remains fixed within an episode. The target convention specifies whether a hint refers to the leftmost or rightmost marked card. The intent convention specifies whether color hints mean PLAY and rank hints mean DISCARD, or vice versa. On each received hint, the receiver then makes two decisions: a target decision, which selects the marked card intended by the sender, and an intent decision, which selects PLAY or DISCARD. We evaluate the two decisions separately. Each player knows its own convention but is not explicitly told the partner's convention.

When A sends a hint to B, its meaning is determined by θ_A, which B must infer from interaction history. The roles reverse when B sends a hint to A. Interpreting a received hint therefore requires combining the sender's convention with the marked slots and hint type, as Fig. 2 (a) illustrates for a single hint. Both plays and discards reveal cards that can help the receiver assess earlier hints, although individual events may remain ambiguous. In particular, a successful play does not by itself establish that the receiver selected the card intended by the sender.

3.2 Interaction Settings

We use scripted interactions between bots and evaluate receiving interventions in controlled games. Each scripted bot sends hints according to its assigned convention and interprets received hints using that same convention. Because the two bots' conventions are drawn independently, the resulting histories contain both reactions that match the sender's intended meaning and reactions that do not. In the controlled games, hint generation is scripted for both players, while, as illustrated in Fig. 2(b),

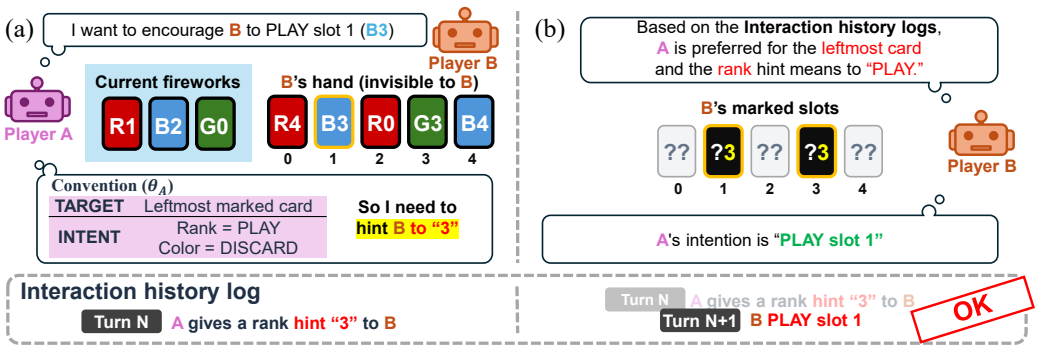


Figure 2: Sender-specific communication conventions. (a) Player A wants B to play slot 1 and, under its target convention (leftmost marked card) and intent convention (rank means PLAY), gives the rank hint 3, which marks slots 1 and 3. (b) Player B sees only the interaction log, infers A's convention from earlier hints and reactions, and interprets the hint as PLAY on slot 1.

LLMs make two receiving decisions at designated turns, selecting the target slot and choosing PLAY or DISCARD. The surrounding policy controls turn selection and actions outside these decisions. We keep this policy and each player's assigned convention fixed across receiving interventions. This setting isolates receiving decisions rather than evaluating fully autonomous game play.

3.3 Model Interface

The base language model's weights remain fixed throughout evaluation. Each prompt contains the task rules, the acting player's own convention, the interaction history, and the game state visible to that player. A NOTES block is present in every condition and contains additional information only when provided by the receiving condition. Target questions offer the distinct leftmost and rightmost marked slots, while intent questions offer PLAY and DISCARD for a specified slot. The two options are labeled A and B, and we select the label with the higher next-token logit at the answer position. In game evaluations, the option order is randomized across decisions. Data partitions and prompt templates are provided in Appendices B and C.

4 Probing and Interventions

Our approach combines linear probes of the partner's convention with textual and activation-based interventions on the receiver's decisions. Oracle and action-translation conditions provide diagnostic comparisons that vary information accuracy and the processing performed outside the model.

4.1 Convention Probes

For an interaction from sender A to receiver B, we predict A's assigned convention from hidden representations of B's observation. At token position t, we concatenate the embedding output and the hidden states from all transformer layers, following work that combines layers to read information distributed across depth (Devlin et al., 2019; Tenney et al., 2019), and fit a separate linear logistic probe for each convention axis:

$$x_t = [h_t^{(0)}; \ldots; h_t^{(L)}], \qquad p_a(\theta_A^a = 1 \mid x_t) = \sigma(w_a^\top x_t + b_a), \tag{1}$$

where $a \in \{\text{target}, \text{intent}\}$ and $h_t^{(0)}$ is the embedding output. Labels identify the sender's convention, independently of the current marked slots and option order. Probes are trained separately for each language model on the scripted interactions of Sec. 3.2, pooling examples from both players while preserving each receiver's observation limits.

Our primary readout, *last-token*, uses the final token of the prompt body, before the action question and options are appended, so it reads the observation provided for action selection without a question eliciting the partner's convention. We use it for both axes and for the readout-guided interventions below. As an auxiliary analysis, the *event-aggregated* readout combines predictions from the final tokens of the history lines that record hints received from the partner and actions taken by the receiver. The selection rule and the training settings are in Appendix C.

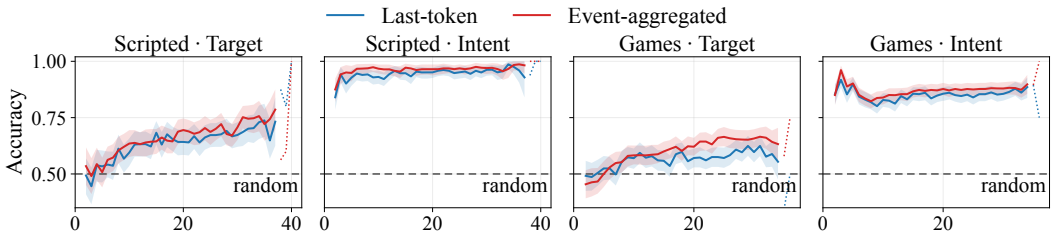


Figure 3: Probe accuracy by turn for Qwen3-8B. Bands are 95% confidence intervals over games, and a curve turns dotted where few games remain.

4.2 Convention Information and Action Selection

We vary two factors: the source of convention information and whether it is translated into the current target and intent decisions. Prediction-based conditions use the last-token probe's estimate of the sender's convention, whereas oracle conditions use the sender's assigned convention.

In `loop`, the probe-predicted target and intent conventions are fed back into NOTES as general rules. The model must then apply these rules to the current hint to make a target decision and an intent decision. The oracle counterpart, `told`, uses the sender's assigned convention in the same rule-statement form. The suffix `verb` denotes action translation. In `loop + verb`, external code applies the probe-predicted convention to the current hint and provides the resulting target slot and intent decision in NOTES. Likewise, `told + verb` applies the sender's assigned convention to the current hint. For example, if a rank hint marks slots 1 and 3 and the supplied convention specifies the leftmost marked card and rank as PLAY, `loop` states those two rules, whereas `loop + verb` recommends slot 1 and PLAY. The LLM still makes the final choice in all conditions.

The `raw` condition receives no partner-convention information. Thus, the oracle conditions vary convention accuracy, while the `+ verb` conditions reduce the convention-to-decision mapping left to the model. The exact wording is given in Appendix C.

4.3 Activation Interventions

We compare donor activation transfers with transformations of the receiver's representations. Donor runs either provide oracle action information or restate the current hint alongside the predicted convention. We transfer attention-head outputs at the answer position and separately test donor attention weights with recipient value vectors (Zhang & Nanda, 2024).

We also evaluate several transformations of the receiver's activations, including head-output scaling, convention-contrast injection, fitted linear maps, and averaged restatement directions (Rimsky et al., 2024). Additional diagnostics test attention to the current hint, query replacement, and a minimal-context self-donor. Matched comparisons hold the state, queried slot, and answer options fixed. The base LLM remains frozen except in the learning diagnostic of Sec. 5.7. Intervention details and implementation checks are in Appendix H.

5 Results

5.1 Models and Evaluation

We evaluate eight open-weight LLMs: Qwen3-8B and Qwen3-32B (Yang et al., 2025), R1-Distill-Qwen-7B (DeepSeek-AI, 2025), Llama-3.1-8B (Grattafiori et al., 2024), Tulu-3-8B (Lambert et al., 2025), Hermes-3-8B (Teknium et al., 2024), Gemma-3-12B (Gemma Team, 2025), and Phi-4 (Abdin et al., 2024). We abbreviate R1-Distill-Qwen-7B as R1-7B in figures and tables. A separate convention probe is trained for each model. We develop the experimental setup on Qwen3-8B and evaluate the probe readouts and prompt-level conditions on all eight LLMs. The matched-state and activation analyses focus on Qwen3-8B. We report *knowing*, *doing*, and game score separately. Game-level comparisons use matched seeds and convention configurations, while state-level analyses compare interventions on identical recorded receiving states.

Table 1: Knowing across eight LLMs, the probe accuracy for the sender's target and intent conventions in scripted interactions and controlled games. Last and Event are the last-token and event-aggregated readouts, Turn-avg. averages over turns, Final uses the last turn of each game.

		Target				Intent			
		Scripted		Games		Scripted		Games	
Model	Readout	Turn-avg.	Final	Turn-avg.	Final	Turn-avg.	Final	Turn-avg.	Final
Qwen family									
Qwen3-8B	Last	0.638	0.677	0.569	0.596	0.921	0.965	0.814	0.873
	Event	0.659	0.747	0.607	0.658	0.964	0.972	0.869	0.883
Qwen3-32B	Last	0.620	0.691	0.564	0.623	0.916	0.927	0.813	0.840
	Event	0.655	0.736	0.585	0.637	0.970	0.990	0.892	0.925
R1-7B	Last	0.612	0.604	0.499	0.506	0.867	0.885	0.782	0.796
	Event	0.637	0.726	0.539	0.588	0.963	0.976	0.883	0.894
Llama family									
Llama-3.1-8B	Last	0.632	0.653	0.504	0.498	0.911	0.955	0.795	0.819
	Event	0.658	0.708	0.534	0.577	0.961	0.979	0.830	0.831
Tulu-3-8B	Last	0.627	0.656	0.500	0.521	0.911	0.944	0.794	0.823
	Event	0.638	0.708	0.549	0.594	0.970	0.983	0.838	0.842
Hermes-3-8B	Last	0.616	0.625	0.499	0.535	0.909	0.948	0.815	0.835
	Event	0.660	0.705	0.538	0.592	0.964	0.972	0.879	0.892
Others									
Gemma-3-12B	Last	0.649	0.688	0.510	0.523	0.909	0.962	0.822	0.867
	Event	0.666	0.760	0.523	0.548	0.940	0.948	0.844	0.850
Phi-4	Last	0.595	0.646	0.506	0.531	0.907	0.958	0.838	0.875
	Event	0.635	0.705	0.563	0.590	0.959	0.969	0.869	0.879

5.2 KNOWING

We first measure knowing, the accuracy with which a probe recovers the sender's assigned convention from the receiver's hidden states, for every LLM on scripted interactions and in controlled games, and report it in Tab. 1. The last-token readout is our primary measure and the event-aggregated readout an auxiliary comparison, and for each axis of θ they predict

$$\hat{\theta}_{\text{last}} = \operatorname{sign} f_{\text{last}}(h_{\text{last}}), \qquad \hat{\theta}_{\text{event}} = \operatorname{sign} \sum_{i \in E_t} f_{\text{event}}(h_i), \tag{2}$$

where h_{last} is the hidden state at the final token of the prompt body, E_t the history lines up to turn t that record hints the partner gave and actions the receiver took, h_i the hidden state at the final token of line i, and f_{last} and f_{event} probe logits trained at those positions. Across LLMs, intent is more accurately decoded than target in both settings. For Qwen3-8B in controlled games, turn-averaged last-token accuracy is 0.569 for target and 0.814 for intent, compared with 0.607 and 0.869 for the event-aggregated readout. Because the readouts differ in token position, event selection, and aggregation, their accuracy difference does not isolate the contribution of external aggregation.

We also track the same accuracy over turns for Qwen3-8B in Fig. 3. Target accuracy generally rises over turns, whereas intent accuracy is high early in the interaction. Later turns contain different board states and fewer surviving games, so each point is the accuracy over the games that reach that turn, not the accuracy of the same games measured at an earlier and a later turn. The per-turn counts and the availability of event-aggregated predictions are in Appendix E. These results show that convention information is accessible to the trained probes, but do not establish that the model uses it when selecting actions (Hewitt & Liang, 2019; Belinkov, 2022).

5.3 DOING

We next measure doing, the rate at which the receiver's target and intent decisions agree with the sender's convention. We report it together with game score under the five receiving conditions of Qwen3-8B in Tab. 2. Without convention information (`raw`), target and intent doing are 0.515 and 0.591, and the predicted convention as a rule statement (`loop`) moves them only to 0.531 and 0.612, with a score increase of $+0.17$.

Table 2: Doing and game score for Qwen3-8B under the five receiving conditions. A rule statement leaves its application to the LLM, whereas action translation provides the target slot and intent decision. Ceiling replaces the LLM by code that follows the sender's convention exactly.

Condition	Convention source	Provided information	Doing (target)	Doing (intent)	Score
`raw`	—	—	0.515	0.591	5.63
`loop`	Probe prediction	Rule statement	0.531	0.612	5.80
`loop + verb`		Action translation	0.510	0.760	6.14
`told`	Ground truth	Rule statement	0.821	0.599	6.16
`told + verb`		Action translation	0.962	0.867	7.78
Ceiling	Ground truth	—	1.000	1.000	8.45

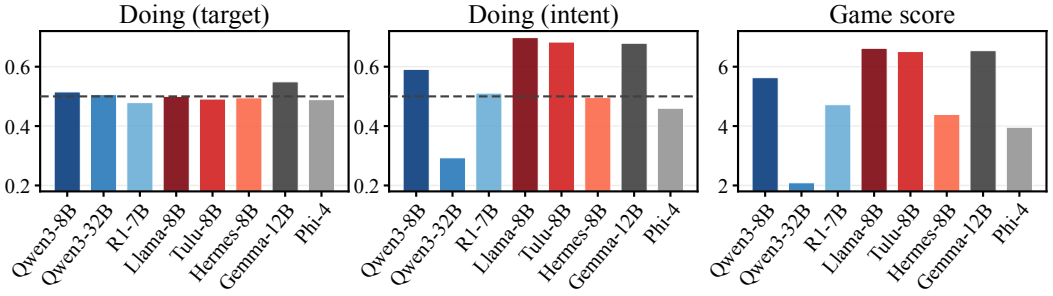


Figure 4: Doing and game score across LLMs with an unassisted receiver (`raw`), higher is better. The dashed line at 0.5 marks random choice. Bar colors mark the model family.

The two axes respond differently to information accuracy and action translation. Applying the same prediction to the current hint (`loop + verb`) raises intent doing from 0.612 to 0.760 while target doing remains at 0.510, with a score increase of $+0.34^*$ over `loop`. Supplying the true convention as a rule statement (`told`) instead raises target doing to 0.821 while intent doing remains at 0.599, with a score increase of $+0.36^*$. Thus, target choices improve with a corrected convention, whereas intent choices respond to its translation into the current target and intent decisions. Combining both (`told + verb`) yields the highest doing on both axes, at 0.962 for target and 0.867 for intent, and a game score of 7.78, $+2.15^*$ over `raw`. Code that follows the sender's convention exactly scores 8.45, so `told + verb` remains 0.67^* below the ceiling of this environment and sending policy. These results show that both information accuracy and action translation affect receiving behavior and cooperation.

5.4 Self-Play and Cross-Play

We then run every LLM with an unassisted receiver and report doing and game score in Fig. 4; the exact values are in Appendix F. We further pair the eight LLMs in self-play and cross-play under the same scripted sending policy and report the scores in Fig. 5. Self-play pairs two instances of the same LLM, while cross-play pairs different LLMs. Each ordered pairing is evaluated on 240 games across four convention configurations.

Across the 56 ordered cross-play pairings, the mean score is 5.1 for `raw`, 5.3 for `loop`, and 5.4 for `told`. Providing rule statements therefore produces relatively small aggregate changes, even when the conventions are correct. In contrast, `told + verb` reaches a cross-play mean of 7.0. Larger gains from action translation also appear in several self-play comparisons. For example, Qwen3-32B scores 2.2 under `raw`, 2.9 under `told`, and 8.4 under `told + verb`. Thus, the effect of translating convention information into the current decision extends beyond Qwen3-8B. Doing and game score under every condition for each LLM are reported in Appendix F.

A similar aggregate pattern appears without ground truth. Translating the predicted convention with `loop + verb` raises the cross-play mean from 5.3 to 5.9. For Qwen3-32B paired with Qwen3-8B, the score rises from 4.2 and 4.1 under `loop` to 6.3 with either model as A. These results show that how convention information is presented to the receiver matters across LLM pairings. Game score alone, however, does not reveal whether the receiver became more sensitive to the convention or mainly shifted its action preference. We examine this distinction with matched-state analyses next.

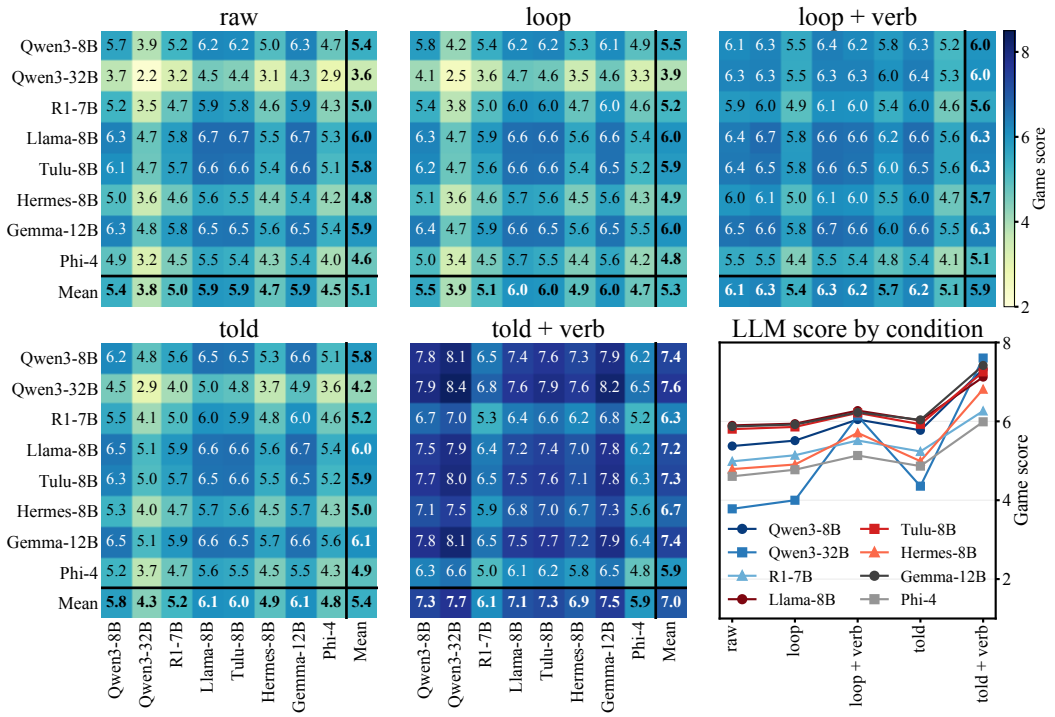


Figure 5: Self-play and cross-play game scores across LLMs and conditions. Rows give the LLM playing as A, which moves first, and columns the LLM playing as B. Mean entries average the eight cells of a row or column. The lower-right panel gives each LLM's mean score under each condition.

Table 3: Sensitivity to statement reversals on 800 matched Qwen3-8B intent decisions. Mean aligned Δ measures the logit shift in the implied direction. Aligned fraction, choice-flip rate, and zero-shift rate report directional shifts, changed choices, and exactly zero shifts, respectively.

Statement	Mean aligned Δ	Aligned fraction	Choice-flip rate	Zero-shift rate
Rule statement (color hint)	0.157	0.91	0.135	0.23
Rule statement (rank hint)	-0.005	0.47	0.080	0.34
Rule statement (all)	0.066	0.68	0.104	0.29
Action recommendation	1.378	1.00	0.586	0.00

5.5 Responses to Convention Information

To separate the two, we reverse the notes statement on 800 recorded receiving states of Qwen3-8B, 348 with a color hint and 452 with a rank hint, while holding the history, the selected slot, the question, and the answer options fixed, once for the rule statement and once for a direct action recommendation, and report the responses in Tab. 3. We focus on the intent axis, where a true rule statement did not change behavior in Tab. 2, and ask whether the receiver responds to the content of the statement at all.

Reversing the action recommendation moves the answer logit by 1.378 on average, always in the implied direction, and changes the selected action in 59% of states. Reversing the rule statement moves it by only 0.066, changes the selection in 10% of states, and leaves the logits unchanged in 29%. Its remaining effect is concentrated in color hints: 91% of moving states shift in the implied direction, versus 47% for rank hints. The receiver therefore follows an explicit action recommendation but applies a stated rule weakly and mainly for one hint type. Aggregate accuracy can hide this distinction: scaling one attention head raises intent accuracy from 0.584 to 0.605 only by choosing PLAY more often, helping PLAY states while harming DISCARD states. We therefore report intervention effects separately for states requiring PLAY and DISCARD; the precision check and scaling results are in Appendix G.

Table 4: Activation-intervention effects on intent accuracy, split by the true intent decision. Changes are against each experiment's matched baseline on its recorded states.

Intervention	Donor	Δ all	Δ PLAY	Δ DISCARD
Output transfer (layer 24)	oracle	+0.091	+0.046	+0.219
Attention transfer (layer 24)	oracle	+0.071	+0.033	+0.184
Restatement transfer (layer 26)	non-oracle	+0.057	+0.051	+0.069
Head scaling (24.29, $\times 1.5$)	none	+0.021	+0.047	−0.053
Attention boost	none	+0.053	+0.100	−0.102
Averaged directions	none	+0.027	+0.054	−0.060
Contrast injection	none	+0.004	+0.002	+0.009
Linear map	none	+0.007	+0.003	+0.018

5.6 EFFECTS OF ACTIVATION INTERVENTIONS

Since a stated rule has little effect on the answer, we ask whether activation interventions can better support the convention-to-decision mapping. We try this in two ways on Qwen3-8B's receiving activations, transferring attention-head outputs from donor runs that provide more explicit decision information, and applying fixed transformations without a donor run, and report the results in Tab. 4.

The best single-layer transfer in a sweep of all 36 layers improves intent accuracy for both action classes: layer 24 for oracle action donors and layer 26 for hint-restatement donors. Transferring oracle attention weights alone also helps. Fixed transformations do not reproduce this benefit. Contrast injection and linear maps change accuracy by less than 0.01, while head scaling, the attention boost, and averaged directions raise PLAY accuracy but lower DISCARD accuracy, shifting action preference. Query replacement and a minimal-context self-donor likewise fail. Linear readouts of attention-block outputs at the answer position do not recover the convention-consistent intent decision from the `loop` run at any layer (0.58–0.63 versus a majority rate of 0.767), whereas they do from the oracle action run from layer 7 (0.966). The hint type and predicted convention are decodable from `loop` at 0.92 and 0.86; combining these readouts outside the model reaches 0.81. The tested readouts thus recover the hint type and predicted convention more reliably than the resulting intent decision at these attention-block outputs. Donor definitions, intervention settings, and detailed results appear in Appendix H.

5.7 LEARNING THE APPLICATION STEP

Since the missing combination is not recovered by the tested fixed interventions, we ask whether it can be learned. We fine-tune a LoRA adapter on Qwen3-8B using 4,000 receiving decisions from the probe training boards, where NOTES contains a randomly sampled rule statement and the label is the target slot and intent decision computed from that rule and the current hint. Neither ground-truth conventions nor probe predictions are used for training. The adapter applies stated rules almost perfectly: under `told`, intent doing rises from 0.599 to 0.998 and game score from 6.16 to 8.43, near the 8.45 ceiling. It also follows incorrect rule statements, indicating that it has learned the rule-application step itself. With the original probe, however, `loop` falls to 4.98 as probe accuracy degrades on the adapter's hidden states. Retraining the probe raises the score to 6.47, and omitting the poorly decoded target statement raises it further to 6.72. The remaining limitation therefore shifts from applying the convention to recovering it. The details are in Appendix I.

6 CONCLUSION

We studied how LLMs infer and use a partner's communication convention when responding to hints. Linear probes decoded the convention from receiver representations, but decisions did not follow it. Stating it as a rule had limited effect; linking it to the current decision helped more, though probe errors bounded the gains. Convention readout, behavioral adaptation, and game score therefore require separate evaluation. Using inferred partner information in decisions remains an open challenge for cooperation with unfamiliar partners.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, implement methods, assist with translation, support qualitative and thematic data analysis, brainstorm research directions, draft parts of the paper, improve readability, and identify relevant literature. We did not use generative AI tools to generate synthetic datasets, develop theoretical or conceptual frameworks, propose or refine hypotheses, clean or reformat datasets, interpret experimental results, formulate mathematical claims, provide critical ingredients for mathematical proofs, or assist in writing proofs. All AI-assisted work was reviewed by the authors. We manually verified AI-suggested related work, reviewed and tested all LLM-generated code for correctness, and checked all AI-assisted text line by line. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We document every component needed to reproduce our results in the main text and the appendix. The eight evaluated LLMs are listed in Appendix A, together with the Hugging Face repository each checkpoint is downloaded from, the commit revision it is pinned to, its architecture and parameter count, and the library versions and loading settings we use. The seed formula of eq. (3), the disjoint board ranges used for probe training, probe testing, constant selection, and the main games, and the game parameters are in Appendix B. Section 3 describes the environment and the scripted bots, the prompt format, the answer interface, and the notes wording of every condition are in Sec. 3.3 and Appendix C, and complete prompts are reproduced verbatim in Appendix D. The probe inputs, the event-selection and aggregation rules, and the probe-training settings are in Sec. 4.1 and Appendix C, and Sec. 4.2 defines every compared condition. All LLMs are used with frozen weights, answers are read from next-token logits without sampling, and the option order at each decision is drawn from a generator seeded by the board seed.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report, 2024.

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.

Raghu Arghal, Fade Chen, Niall Dalton, Evgenii Kortukov, Calum McNamara, Angelos Nalmpantis, Moksh Nirvaan, Gabriele Sarti, and Mario Giulianelli. A behavioural and representational evaluation of goal-directedness in language model agents. In *International Conference on Machine Learning (ICML)*, 2026.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. doi: 10.1016/j.artint.2019.103216.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022. doi: 10.1162/coli_a_00422.

Siddharth Boppana, Annabel Ma, Max Loeffler, Raphael Sarfati, Eric Bigelow, Atticus Geiger, Owen Lewis, and Jack Merullo. Reasoning theater: Disentangling model beliefs from chain-of-thought, 2026.

Joschka Braun, Carsten Eickhoff, and Seyed Ali Bahrainian. Beyond multiple choice: Evaluating steering vectors for summarization. In *Findings of the Association for Computational Linguistics: EACL 2026*. Association for Computational Linguistics, 2026.

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-AI coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. Answer matching outperforms multiple choice for language model evaluation, 2025.

Logan Cross, Violet Xiang, Agam Bhatia, Daniel L. K. Yamins, and Nick Haber. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2407.07086.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Tin Dizdarević, Ravi Hammond, Tobias Gessler, Anisoara Calinescu, Jonathan Cook, Matteo Gallici, Andrei Lupu, and Jakob Nicolaus Foerster. Ad-hoc human-AI coordination challenge. In *International Conference on Machine Learning (ICML)*, 2025.

Gemma Team. Gemma 3 technical report, 2025.

Harsh Goel, Aditya Sai Ellendula, Vaishnav Tadiparthi, Ehsan Moradi Pari, Hossein Nourkhiz Mahjoub, and Sandeep P. Chinchali. Bayesian partner modelling enables adaptive replanning for LLM coordination, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models, 2024.

Tal Haklay, Hadas Orgad, David Bau, Aaron Mueller, and Yonatan Belinkov. Position-aware automatic circuit discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2792–2817, 2025. doi: 10.18653/v1/2025.acl-long.141. URL https://aclanthology.org/2025.acl-long.141/.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275.

Hengyuan Hu, Adam Lerer, Alexander Peysakhovich, and Jakob Foerster. "Other-Play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*. PMLR, 2020.

Bingyu Hui, Lebin Yu, Quanming Yao, Yunpeng Qu, Xudong Zhang, and Jian Wang. Efficient reinforcement learning for zero-shot coordination in evolving games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22110–22118, 2026.

Shawn Im and Sharon Li. A unified understanding and evaluation of steering methods, 2025.

Athul Paul Jacob, David J. Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with KL-regularized search. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *International Conference on Machine Learning (ICML)*, 2025.

Adam Kamel, Tanish Rastogi, Michael Ma, Kailash Ranganathan, and Kevin Zhu. Emergent world beliefs: Exploring transformers in stochastic games, 2025. Accepted at the NeurIPS 2025 Mechanistic Interpretability Workshop (per arXiv listing).

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling (COLM)*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, et al. Tülu 3: Pushing frontiers in open language model post-training, 2025.

Paraskevas V. Lekeas and Giorgos Stamatopoulos. What suppresses Nash equilibrium play in large language models? mechanistic evidence and causal control, 2026.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023a. arXiv:2310.10701.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023b. Oral presentation. arXiv:2210.13382.

Mengfan Li, Xuanhua Shi, and Yang Deng. CoSToM: Causal-oriented steering for intrinsic theory-of-mind alignment in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9302–9317, 2026a. URL https://aclanthology.org/2026.acl-long.421/.

Wenkai Li, Fan Yang, Ananya Hazarika, Shaunak A. Mehta, and Koichi Onoue. When reasoning traces become performative: Step-level evidence that chain-of-thought is an imperfect oversight channel, 2026b.

Fangzhou Liang, Tianshi Zheng, Chunkit Chan, Yauwai Yim, and Yangqiu Song. LLM-Hanabi: Evaluating multi-agent gameplays with theory-of-mind and rationale inference in imperfect information collaboration game, 2025. Accepted to the Wordplay Workshop at EMNLP 2025.

Maxime Méloux, François Portet, and Maxime Peyrard. Mechanistic interpretability as statistical estimation: A variance analysis, 2025.

Meta Fundamental AI Research Diplomacy Team (FAIR), Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

Ruaridh Mon-Williams, Max Taylor-Davies, Elizabeth Mieczkowski, Natalia Vélez, Neil Bramley, Yanwei Wang, Tom Griffiths, and Christopher G Lucas. Partner modelling emerges in recurrent agents (but only when it matters). In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 7687–7713. Curran Associates, Inc., 2025. doi: 10.52202/085713-0263. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0b8e8bfc40184226888e821620b216c9-Paper-Conference.pdf.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2.

Matthew Nguyen, Kyle Cox, Austin Meek, and Iván Arcuschin. On the generalization of steering vectors for chain-of-thought faithfulness, 2026.

Erik Nordby, Tasha Pais, and Aviel Parrack. Linear probe accuracy scales with model size and benefits from multi-layer ensembling, 2026.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Rott Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs. In *International Conference on Learning Representations (ICLR)*, 2026.

Mahesh Ramesh, Kaousheik Jayakumar, Aswinkumar Ramkumar, Pavan Thodima, Aniket Rege, and Emmanouil-Vasileios Vlatakis-Gkaragkounis. Sparks of cooperative reasoning: LLMs as strategic Hanabi agents. In *International Conference on Machine Learning (ICML)*, 2026.

Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D. Weisz, and Murray Campbell. Position: Theory of Mind benchmarks are broken for large language models. In *International Conference on Machine Learning (ICML)*, 2025. arXiv:2412.19726.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Constantin Ruhdorfer, Matteo Bortoletto, Johannes Forkel, Jakob Foerster, and Andreas Bulling. The yōkai learning environment: Tracking beliefs over space and time. In *Reinforcement Learning Conference (RLC)*, 2026.

Raphaël Sarfati, Eric Bigelow, Daniel Wurgaft, Siddharth Boppana, Jack Merullo, Atticus Geiger, Owen Lewis, Tom McGrath, and Ekdeep Singh Lubana. The shape of beliefs: Geometry, dynamics, and interventions along representation manifolds of language models' posteriors, 2026.

Wei Sheng and Rohan Paleja. Beyond partner diversity: An influence-based team steering framework for zero-shot human-machine teaming, 2026.

Shrutika Singh, Anton Alyakin, Daniel Alexander Alber, Jaden Stryker, Ai Phuong S. Tong, Karl Sangwon, Nicolas Goff, Mathew De La Paz, Miguel Hernandez-Rovira, Ki Yun Park, Eric Claude Leuthardt, and Eric Karl Oermann. The pitfalls of multiple-choice questions in generative AI and medical education. *Scientific Reports*, 15(42096), 2025. doi: 10.1038/s41598-025-26036-7.

Johnathan Sun and Andrew Zhang. Persona vectors in games: Measuring and steering strategies via activation vectors, 2026.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. Analysing the generalisation and reliability of steering vectors. *Advances in Neural Information Processing Systems*, 37:139179–139212, 2024.

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. Hermes 3 technical report, 2024.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4593–4601, 2019.

Kale-ab Tessera, Leonard Hinckeldey, Riccardo Zamboni, David Abel, and Amos Storkey. Probing Dec-POMDP reasoning in cooperative MARL. In *Proceedings of the 25th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2026)*, 2026.

Marissa A. Weis, Maciej Wołczyk, Rajai Nasser, Rif A. Saurous, Blaise Agüera y Arcas, João Sacramento, and Alexander Meulemans. Multi-agent cooperation through in-context co-player inference, 2026.

Frank Zhengqing Wu, Francesco Tonin, and Volkan Cevher. Demystifying variance in circuit discovery of LLMs, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report, 2025.

Jianzhu Yao, Kevin Wang, Ryan Hsieh, Haisu Zhou, Tianqing Zou, Zerui Cheng, Zhangyang Wang, and Pramod Viswanath. SPIN-Bench: How well do LLMs plan strategically and reason socially?, 2025.

Richard J. Young. Why models know but don't say: Chain-of-thought faithfulness divergence between thinking tokens and answers in open-weight reasoning models, 2026.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. ProAgent: Building proactive cooperative agents with large language models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2024a. arXiv:2308.11339.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Hf17y6u9BC.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B. Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *International Conference on Learning Representations (ICLR)*, 2024b. arXiv:2307.02485.

Table 5: Architecture of the evaluated LLMs. Heads give the query and key-value head counts, parameters count every weight in the released checkpoint, and the probe input is the length of the concatenated all-layer state at one token position.

Model	Architecture	Layers	Hidden size	Heads (Q/KV)	Parameters	Probe input
Qwen3-8B	Qwen3	36	4096	32/8	8.19B	151,552
Qwen3-32B	Qwen3	64	5120	64/8	32.76B	332,800
R1-7B	Qwen2	28	3584	28/4	7.62B	103,936
Llama-3.1-8B	Llama	32	4096	32/8	8.03B	135,168
Tulu-3-8B	Llama	32	4096	32/8	8.03B	135,168
Hermes-3-8B	Llama	32	4096	32/8	8.03B	135,168
Gemma-3-12B	Gemma3	48	3840	16/8	12.19B[†]	188,160
Phi-4	Phi3	40	5120	40/10	14.66B	209,920

[†]11.77B in the language decoder, the rest in the unused vision tower.

Table 6: Source of the evaluated LLMs. Each checkpoint is pinned to a Hugging Face repository and the first eight characters of its commit hash.

Model	Hugging Face repository	Revision
Qwen3-8B	`Qwen/Qwen3-8B`	`b968826d`
Qwen3-32B	`Qwen/Qwen3-32B`	`9216db57`
R1-7B	`deepseek-ai/DeepSeek-R1-Distill-Qwen-7B`	`916b56a4`
Llama-3.1-8B	`NousResearch/Meta-Llama-3.1-8B-Instruct`	`d10aef79`
Tulu-3-8B	`allenai/Llama-3.1-Tulu-3-8B`	`66694379`
Hermes-3-8B	`NousResearch/Hermes-3-Llama-3.1-8B`	`896ea440`
Gemma-3-12B	`google/gemma-3-12b-it`	`96b6f1ec`
Phi-4	`microsoft/phi-4`	`2db69c1c`

A Evaluated Models

We evaluate eight open-weight instruction-tuned LLMs, and we list their architecture in Tab. 5 and their source in Tab. 6 so that a reader can load the same weights. Every model is loaded from its Hugging Face repository with the Transformers library (version 5.15.1) on PyTorch 2.6.0 with eager attention, and without a beginning-of-sequence token prepended to the prompt. For Llama-3.1-8B we load the NousResearch mirror, whose weights are identical to the gated Meta release. Gemma-3-12B is distributed as a vision-language model, and we use only its language decoder, so the vision tower never receives input.

The probe input at one token position concatenates the residual stream of every layer, counting the embedding output as layer 0, so its dimension is the number of layers plus one times the hidden size.

B Environment and Measurement Constants

We fixed every constant in this appendix before measurement, and we report them here so that a reader can reproduce any number in the main text.

Episodes are organized into disjoint boards by the seed formula

$$\text{seed} = 2{,}300{,}000 + c \times 100{,}000 + i, \tag{3}$$

where c indexes the four convention configurations and i indexes the board. The four configurations enumerate the convention family, leftmost or rightmost target crossed with color or rank as the play signal, so a board index appears once per configuration and every reported cell averages over all four. The agent's own convention in the controlled games comes from a separate function of the same seed, using bits that are independent of the partner's convention, so the agent's own convention carries no information about the convention it has to infer.

The board ranges in Tab. 7 never overlap. We train probes on the training boards, report every readout number on the test boards, choose every readout constant and every game-level threshold on the constant-selection boards, and run every reported game comparison on the main stage boards.

Table 7: Design constants, fixed before measurement. The seed formula of eq. (3) makes the board ranges disjoint.

Probe training boards	indices 0–59 per configuration
Probe test boards	indices 500–535 per configuration
Main stage boards	indices 3000–3059 per configuration (240 games)
Constant-selection boards	indices 2000–2059 per configuration
Convention family	2×2 (target $\times$ intent), drawn per episode
Probe	linear logistic on all layers concatenated at one token position
Score scale	0–15, paired seeds, bootstrap 95% intervals

The matched-state analyses of Secs. 5.5 and 5.6 use recorded receiving states from the main stage boards and from the constant-selection boards, as listed in Appendix H.

The environment is a three-color, five-rank variant, so a game scores from 0 to 15 points. We pair every score difference on the seed, which removes the variance that comes from the deal, and we compute bootstrap 95% confidence intervals by resampling games rather than decisions, because decisions within one game are not independent.

For the code-based ceiling of Tab. 2, the receiving LLM is replaced by a deterministic receiver that applies the sender's assigned target and intent conventions to each received hint. The scripted sending policy and all other game mechanics remain unchanged. The resulting score is therefore a ceiling for receiving under the fixed sending policy, not the maximum game score of 15.

C Prompt and Probe Details

C.1 Prompt Format and NOTES

A decision prompt has four fixed parts, the rules and the agent's own sending convention, the interaction log, a notes block, and the current board, followed by the question and the two options. Every line is serialized to a width fixed by the game configuration rather than by its content, so a log line can be replaced by a blank line of the same width. Fixed-width serialization does not by itself guarantee token alignment, so for every comparison that depends on matched positions we tokenize both prompts and verify that they agree token for token outside the designed difference. The notes block is present in every condition and holds blank filler when a condition writes nothing into it, which is what lets us compare a written sentence against its absence at matched positions.

The readout input is the prompt body that the action prompt extends, so the probe reads a state that the model computes on the way to its answer, although not the answer position itself. A variant that appends a reading question to the prompt raises target accuracy to 0.753 on scripted interactions and 0.617 in games, but it is read from a state the model never answers from, so we do not use it.

Every condition other than `raw` writes four lines into the notes block, and the remaining notes lines keep their blank filler. The rule-statement conditions, `loop` and `told`, write two lines per axis.

Rule statement (`loop` and `told`)

```
- When P1's hint marks more than one of your cards,
- the one P1 means is the leftmost of the marked ones
- When P1 gives you a rank hint, P1 wants you to play
- the card it means; a colour hint means discard it
```

The statement names neither the current hint nor a slot, so the model has to apply it to the hint it has just received. The action-translation conditions, `loop + verb` and `told + verb`, write these four lines in place of the rule statement.

Action translation (`loop + verb` and `told + verb`)

```
- Of the cards this hint marked, the one P1 means
- is the leftmost of them, which is slot 1
- For this hint, what P1 wants you to do with
- that card is to play it
```

Here external code applies the convention to the current hint and fills in the end, the slot number, and the action. In both forms, the values come from the probe predictions in `loop` and `loop + verb` and from the sender's assigned convention in `told` and `told + verb`, and the left end, the rank-play mapping, and the action above are replaced by their alternatives when the convention points the other way. The boxes quote the notes lines as they appear in the prompt without their fixed-width padding, keeping the British spelling of the prompt text.

C.2 Answer Interface

We read answers through two interfaces. The single-token interface takes the answer as the logit difference between the two option tokens at a fixed position of one forward pass, without sampling and without parsing, and the option order is counterbalanced within every pair. The generation interface wraps the prompt in the model's chat template, lets the model write freely, and parses a final line into a choice.

Every number in the main text comes from the single-token interface. Probing and patching take the state at a matched token position as part of their definition, and under free generation the model-written prefix differs across samples, so the matched position that these measurements require does not exist there. We therefore use the generation interface only as a check, packaging the same decisions for free generation and repeating the readout, and we report numbers from the two interfaces side by side rather than pooling them, because the chat template and the generated prefix change what the model conditions on.

C.3 Event-Aggregated Readout

The event-aggregated readout selects the received-side lines of the history, which are the hints the partner gave to this receiver and the actions this receiver took. Lines recording hints this receiver sent, and the partner's reactions to them, are excluded, because those lines follow this receiver's own convention rather than the partner's. Selection uses only observable records, never the sender's true convention or unrevealed cards. At a given turn, we sum the probe logits over the selected lines up to that turn and predict the convention by the sign of the sum. An individual line need not identify the convention, and because the line representations share preceding context, we do not treat their predictions as independent evidence. When no line is selected, as at the first turn, the event-aggregated readout makes no prediction.

C.4 Probe Training

Each probe standardizes every feature with the mean and standard deviation of the training set and fits an L_2-regularized logistic regression, with inverse regularization strength $C = 0.01$, by L-BFGS for at most 300 iterations.

D Prompt Examples

We reproduce two prompts verbatim, both written from the receiver's perspective, in which the receiver is always P0 and its partner P1. The first box is the readout input at turn 12 of a scripted interaction on probe training board 0 of the first configuration, where the partner's convention is the rightmost target with rank hints meaning play, and the probe reads its final token. The second box is a receiving decision at turn 18 of a raw game played by Qwen3-8B on board 3000 of the same configuration, where the partner's convention is the leftmost target with rank hints meaning play. It omits the rule text shared with the first box and shows the target question and the action question, each asked in its own forward pass on the same prompt body. The notes block reads none in both

prompts because neither setting writes into it, and the trailing dots are fixed-width padding. The model picks slot 1, the leftmost marked card as the partner's convention implies, and then plays it on a color hint, as its own convention implies rather than the partner's. The remaining conditions use the same prompt body and differ only in the NOTES lines of Appendix C.

Scripted interaction, readout input at turn 12

```
This game is played by P0 and P1. You are P0.
No player can see their own cards; every player sees every other player's hand.
Your hand has 5 slots, numbered 0 to 4 from left to right.
A card is playable if its rank is exactly one above the current stack of its colour.
Playing a playable card advances that colour's stack. Playing an unplayable card
costs a life.
When you play or discard the card in slot k, that card leaves your hand, every card
to its
right moves one slot to the left, and a freshly drawn card takes the rightmost slot.
So a
slot number names a position in your hand, not a fixed card.
A hint names one colour or one rank and marks every card of the receiver's hand that
matches.
Cards that are not marked are known not to match.
In the log, marks [0,1,-,-,4] means slots 0, 1 and 4 are marked and slots 2 and 3 are
not.
Near the end of the game, after the deck runs out, hands shrink; `x` marks a slot
that no
longer exists, which is different from `-`, a slot whose card simply does not match
the hint.
Where card ids appear, each card keeps its id for as long as it stays in the hand, so
ids let
you follow a card across slot changes; ids reveal nothing about a card's colour or
rank.
LOG LEGEND. Each line is one action. T07 is turn number 7; turns go around the
players in
order. Colours are R (red), Y (yellow), G (green). A card is written as its colour
letter and rank, so Y1 is the yellow 1. Verbs: hint = give a hint; play = play the
card in a slot; disc = discard the card in a slot. After play, -> OK means the card
was playable and advanced its stack, -> BAD means it was not and cost a life. A
discard line shows -> -- because a discard has no success or failure. Trailing dots
are padding with no meaning.
Each partner is assigned rules of its own that are consistent, never stated in the
game log, and never change
within a game. Different partners may use different rules.
When a partner gives you a hint, which of the marked cards it means follows that
partner's rule, and whether it asks you to play or to discard that card depends on
the hint being a colour or a rank hint, per that partner's rule.
When you give a partner a hint, it is meant to trust it and act on it on its next
turn, whenever
that move is legal: which marked card it takes you to mean, and whether it plays or
discards that card depending on your hint being a colour or a rank hint, also follow
that partner's rules.
Infer each partner's rules from that partner's own hints and responses so far.
A hint has two sides: what its sender means by it follows the sender's rules; what
its receiver takes it to mean under its own rules follows those rules. The two can
differ.
Your own moves in the log are meant to follow YOUR RULES below. Under them, your
hints use your play kind aimed at your end when that card is playable, or the
other kind at your end when that card is already on the stacks, as a discard
signal; when a partner hints you, you take the end YOUR RULES name and play it
on your play kind or discard it otherwise, whenever legal; your other discards
pick one of your unmarked slots at random (any slot if all are marked), and with
no legal discard you play a random slot. YOUR RULES describe that assigned
habit, not a guarantee that every logged move obeyed it; when you choose now,
weigh how each partner encodes and reads hints by its own rules.
Each hand line in BOARD lists one other player's cards left to right; `xx` is a slot
```

```
that no longer exists.

YOUR RULES
  - When you hint, you mean the leftmost marked card . . . .
  - You signal play with colour hints; rank means discard . . . . .

HISTORY
T01 P1 hint RANK  1 to P0 marks [0,-,2,-,-] . . . . . .
T02 P0 disc slot 0 -> --  card G1 . . . . . .
T03 P1 hint RANK  1 to P0 marks [-,1,-,-,4] . . . . . .
T04 P0 disc slot 1 -> --  card G1 . . . . . .
T05 P1 hint RANK  1 to P0 marks [-,-,-,3,-] . . . . . .
T06 P0 disc slot 3 -> --  card G1 . . . . . .
T07 P1 play slot 4 -> BAD card G5 . . . . . .
T08 P0 play slot 0 -> BAD card R5 . . . . . .
T09 P1 play slot 0 -> BAD card R3 . . . . . .
T10 P0 play slot 3 -> BAD card R2 . . . . . .
T11 P1 hint RANK  1 to P0 marks [-,-,-,-,4] . . . . . .

NOTES
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .

BOARD
stacks R0 Y0 G0 | hints left 7 | lives 11 | deck 13
P1 hand R4 G3 Y5 R3 Y1
```

Raw game with Qwen3-8B, receiving decision at turn 18

```
[Rules and log legend, identical to the first box]

YOUR RULES below. Under them, your
hints use your play kind aimed at your end when that card is playable, or the
other kind at your end when that card is already on the stacks, as a discard
signal; when a partner hints you, you take the end YOUR RULES name and play it
on your play kind or discard it otherwise, whenever legal; your other discards
pick one of your unmarked slots at random (any slot if all are marked), and with
no legal discard you play a random slot. YOUR RULES describe that assigned
habit, not a guarantee that every logged move obeyed it; when you choose now,
weigh how each partner encodes and reads hints by its own rules.
Each hand line in BOARD lists one other player's cards left to right; `xx` is a slot
that no longer exists.

YOUR RULES
  - When you hint, you mean the leftmost marked card . . . .
  - You signal play with colour hints; rank means discard . . . . .

HISTORY
T01 P1 hint RANK  1 to P0 marks [-,-,2,3,-] . . . . . .
T02 P0 play slot 2 -> OK  card G1 . . . . . .
T03 P1 hint RANK  1 to P0 marks [-,-,2,-,-] . . . . . .
T04 P0 play slot 2 -> OK  card R1 . . . . . .
T05 P1 disc slot 1 -> --  card R4 . . . . . .
T06 P0 hint COLOR R to P1 marks [0,-,-,-,-] . . . . . .
T07 P1 play slot 0 -> OK  card R2 . . . . . .
T08 P0 disc slot 0 -> --  card Y2 . . . . . .
T09 P1 hint RANK  2 to P0 marks [-,-,-,3,-] . . . . . .
```

```
T10 P0 play slot 3 -> OK  card G2 . . . . . .
T11 P1 disc slot 2 -> --  card Y1 . . . . . .
T12 P0 hint RANK  1 to P1 marks [-,1,-,-,4] . . . . . .
T13 P1 disc slot 1 -> --  card G1 . . . . . .
T14 P0 disc slot 1 -> --  card Y3 . . . . . .
T15 P1 hint RANK  3 to P0 marks [0,-,-,-,-] . . . . . .
T16 P0 play slot 0 -> OK  card G3 . . . . . .
T17 P1 hint COLOR R to P0 marks [-,1,2,-,-] . . . . . .

NOTES
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .
  - none . . . . . .

BOARD
stacks R2 Y0 G3 | hints left 6 | lives 15 | deck 10
P1 hand Y3 G5 R5 Y1 G3
```

Target question (first forward pass)

```
P1 just marked slots 1,2. Which one of them do you act on?
OPTIONS
A) SLOT 1
B) SLOT 2

Answer with one letter only.
ANSWER:
```

Model answer A (slot 1)

Action question (second forward pass, same body)

```
P1 just marked slots 1,2. Will you play or discard slot 1?
OPTIONS
A) DISCARD SLOT 1
B) PLAY SLOT 1

Answer with one letter only.
ANSWER:
```

Model answer B (play slot 1)

E TURN CURVES

The two readouts do not produce a prediction under the same conditions. The last-token readout produces one at every turn, since the final token of the decision prompt always exists. The event-aggregated readout produces one only when at least one received-side line is present, which never holds at the first turn and holds for half of the perspectives at the second turn, because the perspective that moves first has not yet received anything. Availability reaches 0.94 by the eighth turn, as Tab. 8 shows. Figure 3 therefore restricts both curves to the turns at which both readouts produce a prediction, since reporting the event-aggregated readout over its own turns would compare it against an easier sample.

The final-turn columns of Tab. 1 are not the rightmost points of Fig. 3. Each perspective contributes at its own last turn, and those last turns are spread across the horizontal axis. In controlled games, 150 of 480 perspectives end at turn 33 and 128 end at turn 34, while only 4 reach turn 36, so the rightmost point of the figure averages over those 4 perspectives.

Table 8: Availability of the event-aggregated readout by turn. Each entry is the fraction of perspectives at that turn for which at least one received-side line is present.

Turn	1	2	3	4	6	8
Scripted interactions	0.000	0.500	0.538	0.632	0.833	0.896
Controlled games	0.000	0.500	0.535	0.652	0.810	0.940

We compute every curve on matched observations and average over observations rather than over perspectives. The released tables give, for each turn, the axis, the readout, the sample restriction, the accuracy under both averaging rules, the number of correct predictions, the number of observations, and the numbers of distinct perspectives and games, together with the bootstrap interval.

F Per-Model Doing and Game Score under Every Condition

Tables 9 and 10 give, for every LLM, the game score and doing under the five LLM receiving conditions of Tab. 2, excluding the code-based ceiling, and under the two threshold variants of Appendix H, on the main stage boards with 240 games per condition, and the `raw` rows are the values plotted in Fig. 4. The threshold is the per-model median of the target readout logit magnitude on the constant-selection boards.

G Statement Sensitivity Analyses

G.1 Matched-State Statement Reversals

We evaluate 800 recorded receiving states, including 348 color-hint states and 452 rank-hint states. Within each comparison, we keep the history, the slot selected by the unassisted receiver, the question, and the answer options fixed. The rule-statement conditions differ only in whether color or rank hints mean Play. The action-recommendation conditions instead state that the current hint means Play or Discard. The paired rule statements have equal token lengths and differ at two token positions. The paired action recommendations also have equal lengths and differ at one position. Option A is fixed to Play and option B to Discard in this diagnostic.

Let $z(x) = \ell_A(x) - \ell_B(x)$ denote the next-token logit margin between the two answer labels. We define the convention-aligned change as

$$\Delta_{\text{conv}} = s_t \left[z(x_{\text{color-play}}) - z(x_{\text{rank-play}}) \right], \qquad s_t = \begin{cases} +1, & \text{for a color hint,} \\ -1, & \text{for a rank hint.} \end{cases} \tag{4}$$

A positive value indicates movement in the direction implied by the convention change. The expected direction depends on the current hint type, not on which convention is the ground truth. For the action recommendations, we define

$$\Delta_{\text{act}} = z(x_{\text{play}}) - z(x_{\text{discard}}). \tag{5}$$

Choice changes indicate whether the higher-logit answer differs between the paired statements.

The mean convention-aligned change is 0.066, with a game-level bootstrap 95% confidence interval of $[0.053, 0.079]$. Its mean is 0.157 for color hints and -0.005 for rank hints. The median absolute logit change is 0.125 for rule statements and 1.250 for action recommendations. Action-recommendation changes follow the recommended action in all 800 states. Because this diagnostic fixes the answer order, these results characterize statement sensitivity under that interface rather than performance averaged over option orders.

G.2 Numerical Precision

We check the sensitivity of small convention effects to numerical precision on 120 states. Recomputing the logits in FP32 changes the mean aligned convention effect from 0.1062 to 0.1025 and the action-recommendation effect from 1.4990 to 1.4968. However, the fraction of exactly zero

Table 9: Doing and game score under every condition, Qwen family and others. Chance is 0.500 for doing and the score is on the 0–15 scale, higher is better.

Model	Condition	Score	Doing (target)	Doing (intent)
Qwen3-8B	raw	5.63	0.515	0.591
	loop	5.80	0.531	0.612
	loop + threshold	5.95	0.545	0.629
	loop + verb	6.14	0.510	0.760
	loop + verb + threshold	6.28	0.552	0.753
	told	6.16	0.821	0.599
	told + verb	7.78	0.962	0.867
Qwen3-32B	raw	2.09	0.506	0.293
	loop	2.52	0.519	0.349
	loop + threshold	3.31	0.521	0.427
	loop + verb	6.31	0.517	0.824
	loop + verb + threshold	5.73	0.514	0.730
	told	2.94	0.939	0.402
	told + verb	8.37	0.959	0.985
R1-7B	raw	4.72	0.479	0.511
	loop	4.98	0.500	0.533
	loop + threshold	4.83	0.482	0.521
	loop + verb	4.92	0.480	0.562
	loop + verb + threshold	4.75	0.482	0.541
	told	5.02	0.530	0.535
	told + verb	5.35	0.533	0.591
Gemma-3-12B	raw	6.51	0.547	0.682
	loop	6.54	0.561	0.681
	loop + threshold	6.51	0.555	0.680
	loop + verb	6.56	0.464	0.764
	loop + verb + threshold	6.67	0.504	0.746
	told	6.49	0.610	0.667
	told + verb	7.87	0.965	0.811
Phi-4	raw	3.96	0.489	0.460
	loop	4.18	0.481	0.478
	loop + threshold	4.28	0.483	0.483
	loop + verb	4.08	0.447	0.548
	loop + verb + threshold	4.06	0.470	0.537
	told	4.29	0.511	0.482
	told + verb	4.78	0.714	0.620

convention changes falls from 27.5% to zero. Among the 33 states with zero changes in the lower-precision measurements, the FP32 median absolute change is 0.0426, and 63.6% of changes follow the convention-implied direction. These measurements support the difference in effect magnitude but do not support interpreting lower-precision zeros as complete insensitivity.

H Activation Interventions and Additional Diagnostics

H.1 Scope and Donor Conditions

These experiments analyze the intent decision of Qwen3-8B. Within each matched-state comparison, the recipient and donor use the same receiving state, selected slot, action question, and option order. The base model remains frozen throughout. Ground-truth conventions are used for oracle donors and retrospective evaluation, but not to construct prediction-based interventions.

The recipient run, denoted L, receives the probe-predicted convention as a rule statement. The oracle action donor, denoted A, instead receives the target slot and intent decision obtained by applying the true convention to the current hint. The restatement donor retains prediction-based convention information and restates information about the current hint. It receives neither the ground-truth convention nor an externally computed action.

Table 10: Doing and game score under every condition, Llama family. Chance is 0.500 for doing and the score is on the 0–15 scale, higher is better.

Model	Condition	Score	Doing (target)	Doing (intent)
Llama-3.1-8B	`raw`	6.62	0.500	0.698
	`loop`	6.65	0.498	0.703
	`loop` + threshold	6.68	0.498	0.709
	`loop + verb`	6.62	0.489	0.754
	`loop + verb` + threshold	6.46	0.500	0.743
	`told`	6.59	0.508	0.699
	`told + verb`	7.25	0.588	0.818
Tulu-3-8B	`raw`	6.51	0.491	0.683
	`loop`	6.65	0.494	0.700
	`loop` + threshold	6.57	0.487	0.701
	`loop + verb`	6.54	0.460	0.776
	`loop + verb` + threshold	6.58	0.480	0.767
	`told`	6.60	0.505	0.693
	`told + verb`	7.56	0.708	0.846
Hermes-3-8B	`raw`	4.39	0.495	0.496
	`loop`	4.46	0.507	0.506
	`loop` + threshold	4.39	0.498	0.497
	`loop + verb`	5.48	0.451	0.692
	`loop + verb` + threshold	5.19	0.476	0.639
	`told`	4.49	0.534	0.501
	`told + verb`	6.65	0.859	0.771

The oracle-transfer analysis uses 899 states from boards 3000–3019 for which the donor and recipient inputs have matching token lengths, a condition that excludes no decision there. The development analyses use 926 states from boards 2000–2019. The fitted linear-map result is measured on a 452-state test split within the development data. These state sets are distinct, so their effect sizes should not be interpreted as a ranking under a common evaluation distribution. Layer 24 and the two heads of the contrast injection were selected on the same main stage states that report them, so those values are selection values, whereas the 206-head set, the linear maps, and the restatement transfer were selected on the constant-selection boards. Paired differences carry 95% bootstrap intervals over boards where the record has them. We did not test borrowing the attention of other heads, attention temperature, fixed additive corrections before the softmax, answer prefill, or transferring only the queries or only the keys of a donor run.

H.2 Donor Activation Transfers

We replace the recipient's concatenated attention-head output at the answer position with the corresponding donor output, before the output projection. We transfer each of the 36 layers in turn and report the layer with the largest gain, layer 24 for the oracle donor and layer 26 for the restatement donor, and the layer order is the same on the main stage and the constant-selection boards. Replacing all head outputs at layer 24 with those from the oracle action donor raises intent accuracy from 0.603 to 0.694. The paired change is 0.091, with a bootstrap 95% confidence interval of $[0.072, 0.112]$. The intervention corrects 91 decisions and introduces nine errors. The oracle donor itself has an accuracy of 0.822 on these states. The same layer-24 transfer on the 926 development states gives 0.094, and transferring the 206 heads of seven layers selected by transfer gain reaches 0.814, the level of the oracle donor itself at 0.806. Transferring the oracle rule donor, which writes the true convention as a rule statement, gives 0.010 at layer 24 with interval $[0.001, 0.019]$ and nothing elsewhere.

We also separate the donor's attention weights from its value vectors. Using the oracle donor's attention weights with the recipient's value vectors improves accuracy by 0.071. Restoring the recipient's attention weights at the NOTES positions, with the corresponding normalization, leaves an improvement of 0.069. Thus, increased attention to NOTES is not necessary for most of the improvement observed in this particular transfer comparison. These effects do not establish that

Table 11: Head-output scaling on 2,717 receiving decisions, head 24.29 of Qwen3-8B under the convention loop.

Measurement	Scale 1.0	Scale 1.5
Overall intent accuracy	0.584	0.605
Accuracy on PLAY states	0.694	0.741
Accuracy on DISCARD states	0.248	0.195
Fraction of PLAY choices	0.709	0.757

attention weights are free of action information, since the donor has already received the oracle action recommendation.

The non-oracle restatement donor provides a further positive comparison. Transferring all attention-head outputs at layer 26 improves intent accuracy by 0.057 on the 926 development states, with a bootstrap 95% confidence interval over boards of $[0.035, 0.080]$, and the full restatement run itself improves it by 0.078 with interval $[0.047, 0.108]$. It corrects 69 decisions and introduces 16 errors, with improvements on both action classes. This intervention requires a separate restatement execution for each decision. Its result is therefore distinct from applying a fixed vector estimated across states.

H.3 RECIPIENT-SIDE TRANSFORMATIONS

Head-output scaling. We multiply selected recipient head outputs by a scalar g. For a head output o_h, the intervention is

$$o'_h = g o_h. \tag{6}$$

Experiments include scaling head 24.29 and applying a shared scale to a set of 206 heads selected through donor-transfer analyses. The shared-scale experiment evaluates $g \in \{0.5, 0.7, 1.3, 1.5, 2.0\}$. Its best setting corrects only two additional decisions out of 926. Individual-head scaling can raise aggregate accuracy by favoring PLAY, while reducing accuracy on DISCARD states.

Action preferences under head scaling. We evaluate output scaling of head 24.29 on 2,717 receiving decisions under the convention-loop condition. The reference action is PLAY in 2,044 decisions and DISCARD in 673 decisions. Unlike the statement-reversal diagnostic, this experiment follows the game's randomized option-order procedure and uses the same assigned order across scaling conditions. A scale of 1.0 leaves the head output unchanged.

Increasing the scale benefits decisions requiring PLAY while harming those requiring DISCARD. Since approximately 75.2% of decisions require PLAY, the majority-class improvement outweighs the minority-class loss in aggregate accuracy. This increase therefore does not by itself establish improved application of the partner's convention.

Convention-contrast injection. For each state, we obtain head outputs under the predicted convention and its reversal, denoted o_h^p and o_h^o. The two inputs share the same answer order and differ only in the rule statement. We intervene with

$$o'_h = o_h^p + \alpha_h \left(o_h^p - o_h^o\right). \tag{7}$$

For head 24.29 with $\alpha_h = 1$, intent accuracy changes by 0.004 across 2,717 decisions, with a 95% confidence interval of $[-0.002, 0.010]$. The intervention corrects 41 decisions and introduces 30 errors.

For head 26.25 with $\alpha_h = -1$, the intervention replaces the predicted-convention output with the opposite-convention output. It corrects and harms 39 decisions each. Applying the two interventions together corrects and harms 38 decisions each. Equal aggregate accuracy therefore does not imply that individual decisions remain unchanged. The direction has little to carry, because reversing the rule statement barely moves the answer logit at any layer, as the layer sweep below shows, so the difference between the two head outputs is close to noise.

Fitted linear maps. We fit linear transformations of the recipient's 128-dimensional head outputs to approximate the corresponding oracle donor outputs. We examine maps with and without an

intercept, low-rank variants of ranks 3 and 8, and selected sets of heads. The base LLM weights remain unchanged, although the linear transformations are fitted using development data.

Some maps explain approximately 20–50% of the donor–recipient difference variance on their test split. However, the best tested intervention yields only three additional correct decisions out of 452, corresponding to an accuracy change of approximately 0.007. The reported uncertainty interval includes zero. Reconstructing part of the activation difference therefore does not establish recovery of its behavioral effect.

Averaged restatement directions. We compute the answer-position residual difference between a run with hint restatement and the corresponding convention-loop run. The differences are averaged separately by hint type and applied to other states. We evaluate individual layers between 6 and 30 and a condition that intervenes at five layers together.

The averaged directions do not reproduce the restatement donor's improvements on both action classes. Earlier-layer interventions primarily favor PLAY, whereas later-layer interventions primarily favor DISCARD. Similar directional changes occur when the vectors are applied to the unassisted receiver. The average intervention therefore does not preserve the behavioral effect of transferring the state-specific restatement execution. The combination that the restatement produces is not stored at the answer position as a fixed direction but arises when the restated hint type is read together with the rule statement and the history through attention, so its average over states keeps only the action-preference component.

H.4 ADDITIONAL MECHANISTIC DIAGNOSTICS

Layer sweep. On 160 states we replace the head outputs of one layer in the run that states rank hints mean play with those of the run that states color hints mean play. Reversing the rule statement moves z by 0.105 on average and changes the choice in 13% of states, and no layer carries more than 0.045 of that shift, with layer 26 the largest; injecting large random vectors at the same positions moves z by 1.4 to 2.8, so the hook is not at fault.

Attention boost. We add a constant to the attention logits from the answer position toward the hint-type token of the current hint line, or toward the whole line, before the softmax. The best setting, a boost of 8 at layers 20–30 toward the type token, gives $+0.053$ $[0.031, 0.075]$ on the 926 development states, but it raises the rate of choosing PLAY from 0.722 to 0.823, gains 0.100 on PLAY states and loses 0.102 on DISCARD states, and moves the unassisted receiver in the same direction, while boosting the whole line reduces accuracy. Making the answer position read the hint-type token more is not the same as combining the meaning of that token with the stated rule, so the boost changes how much is read and not what is concluded.

Query replacement. Replacing the query at the answer position with the query of the rule statement or of the hint-type token, at layer 24 or at seven layers, on 45 states of one board, lowers accuracy from 0.689 to between 0.40 and 0.64, indistinguishable from a control that uses the query of the board line.

Self-donor with minimal context. A donor run of the same model that keeps only the current hint line in the history and the same notes block reaches 0.572 against 0.617 for the recipient, and transferring its head outputs reduces accuracy, by -0.016 $[-0.040, 0.007]$ at layer 24 and -0.049 $[-0.081, -0.015]$ over seven layers.

H.5 WHY THE TESTED FIXED TRANSFORMATIONS DO NOT REPRODUCE THE TRANSFER

For the heads that contribute most to the oracle action transfer, the difference between donor and recipient outputs is 10 to 25% of the output size, 79 to 89% of its variance lies in three directions, and along those directions the donor output equals the recipient output with slope 0.96 to 1.05 and no offset. The difference is therefore a state-specific residual rather than a rescaling, and scalar scaling explains none of it under cross-validation. A 128×128 map per head explains 11 to 49% of that variance on held-out boards, and 57 to 59 of 224 heads exceed 0.2, yet the mapped run makes the same choices as the recipient.

Ridge readouts of the attention-block outputs, trained with four-fold cross-validation over boards on the 926 development states against a majority rate of 0.767, decode the convention-consistent intent decision from the recipient run at 0.58 to 0.63 at every layer, and from the oracle action run at 0.966 from layer 7 and 1.000 from layer 19. The intent decision obtained by combining the predicted convention with the hint type outside the model, which agrees with the true one at 0.865, is likewise not decodable from the recipient run, at 0.58 to 0.65. The two premises are decodable from the recipient run, the hint type at 0.92 and the convention at 0.86 at layer 24, combining the two readouts by their agreement gives 0.81, and a direct linear readout of the intent decision gives 0.60, so the premises reach the answer position and their combination does not. An RBF-kernel readout of the intent decision gives 0.76, at the majority rate. These readouts cover attention-block outputs at the answer position and are linear, so they leave open information held in the residual stream or the MLP outputs, at earlier positions, or in nonlinear form.

H.6 Restating the Hint Type

Outside the game prompt, with only the rule statement, the current hint in words, and the question, 40 prompts covering every convention, hint type, slot, and option order, the model answers at 0.975 with the game wording of the rule. Adding the pieces of the game prompt one at a time, the drop to 0.550 comes when the hint is written in the format of a history line, and restating the hint type in the question restores 1.000 with every other piece of the game prompt present. In games on the main stage boards, restating the type of the current hint in the action question raises doing on the intent axis of `loop` from 0.612 to 0.682 and its score from 5.80 to 5.97 $[-0.04, +0.38]$, but lowers the score of `raw` from 5.63 to 5.22 $[-0.63, -0.18]$ by moving its rate of choosing PLAY from 0.730 to 0.634. The gain of the rule statement given the restatement is $+0.75$ $[+0.52, +0.99]$, while the restatement alone is a bias, so we do not use it as a method.

H.7 Confidence Threshold on the Target Axis

On the constant-selection boards the loop follows its target sentence almost regardless of whether it is right, at target doing 0.800 when the readout is right and 0.127 when it is wrong. We therefore tested omitting the target sentence when the magnitude of the readout logit is below its median on those boards, 2.93 for Qwen3-8B, with a per-LLM median for the other LLMs. On the main stage boards this raises the score of `loop` from 5.80 to 5.95 $[-0.05, +0.35]$, which is $+0.33$ $[+0.13, +0.53]$ over `raw`, and that of `loop + verb` from 6.14 to 6.28 $[-0.07, +0.35]$, while on Gemma-3-12B, which does not follow the sentence, it changes nothing. Deriving the threshold from the same quantities on the constant-selection boards gives no threshold, because target doing without any target sentence is 0.414 there, and applying the threshold on those boards lowers target doing from 0.524 to 0.482. We therefore report the threshold variants with the per-model conditions of Appendix F and do not treat the threshold as a method.

I Learning the Application Step

I.1 Training Data and Objective

The adapter is trained on receiving decisions recorded while the base Qwen3-8B plays `raw` self-play on the probe training boards, 240 games over the four configurations. At each decision we write one of the four rule statements, leftmost or rightmost target crossed with color or rank meaning play, drawn at random into the NOTES field in the `loop` wording, and the label is the target slot and intent decision that code derives from that statement and the current hint, so the statement itself is the supervision and neither the assigned convention nor a probe prediction is used. Prompts in the `raw` format, without a statement, are labeled with the base model's own answer so that unassisted behavior is preserved. Of 20,380 recorded lines we use 4,000, 2,000 action and 1,000 slot questions with a statement and 500 of each without.

I.2 Adapter Configuration

The adapter has rank 16 and scaling 32 on every attention and MLP projection, 43.6M trainable parameters, and is trained for one epoch of 500 steps with AdamW at learning rate 10^{-4} and gradient

Table 12: Game score and doing for Qwen3-8B with the application adapter on the main stage boards, 240 games per condition. No target marks `loop` with the target sentence omitted, retrained marks the probe trained on the adapter's hidden states, the last two columns are paired differences from the adapter under `raw` and from the base LLM under `raw`.

				Doing			
Condition	Model	Probe	Score	Target	Intent	Δ adapter `raw`	Δ base `raw`
`raw`	base	—	5.63	0.515	0.591		
`loop`	base	base	5.80	0.531	0.612		+0.17
`loop`, no target	base	base	6.06	0.503	0.645		+0.43
`told`	base	—	6.16	0.821	0.599		+0.53
`told + verb`	base	—	7.78	0.962	0.867		+2.15
`raw`	adapter	—	6.21	0.505	0.657		+0.58
`loop`	adapter	base	4.98	0.573	0.651	−1.23	−0.65
`loop`	adapter	retrained	6.47	0.467	0.852	+0.25	+0.84
`loop`, no target	adapter	base	4.73	0.473	0.603	−1.48	−0.90
`loop`, no target	adapter	retrained	6.72	0.509	0.797	+0.51	+1.10
`told`	adapter	—	8.43	1.000	0.998	+2.21	+2.80

accumulation 8, with a cross-entropy loss over the two option tokens at the answer position, on one GPU in 1 hour 42 minutes. The training loss falls from 1.34 to between 0.10 and 0.19. In games the adapter is merged into the base weights.

I.3 Probe Retraining

The base probe loses accuracy on the adapter's hidden states, from 0.865 to 0.774 on intent over the constant-selection states, so we retrain the probes on scripted interactions replayed through the adapter model, which gives 0.921 on intent and 0.650 on target on the probe test boards. Ground-truth convention labels are used only to train the post-adapter probe and for retrospective scoring, not to train the LoRA adapter.

I.4 Game Results

Table 12 gives the game results on the main stage boards. Because the adapter also changes behavior without a statement, raising `raw` from 5.63 to 6.21, its conditions are compared with the adapter under `raw` as well as with the base LLM under `raw`. Omitting the target sentence means that the NOTES field holds only the intent statement.

On the constant-selection states the adapter follows the statement it is given at 0.998, so its intent doing under `loop` equals the probe's accuracy, and it changes sending behavior in ways we have not measured. The adapter is trained and evaluated on Qwen3-8B only and has not been placed in cross-play.

J Additional Discussion and Limitations

Access is not application. Convention information is decodable (Tab. 1), and a true convention stated as a rule changes the intent decision little (Tab. 2), which points to the step that applies the estimate to the current hint rather than to the estimate itself. The action-translation conditions perform that step outside the model (Sec. 4.2), the donor runs that help carry more explicit decision information (Tab. 4), and the adapter of Sec. 5.7 learns it, so improving the readout alone is unlikely to close the gap.

What counts as adaptation. A rule statement produces weak, hint-dependent responses, whereas an action recommendation moves the answer directly (Tab. 3), and an intervention can raise aggregate accuracy by favoring the more frequent action (Sec. 5.5 and Appendix G). Neither a higher accuracy nor a higher game score by itself shows that the receiver became more sensitive to its partner, which is why we report knowing, doing, and game score separately (Sec. 5.1) and split intervention effects by states requiring PLAY and DISCARD (Tab. 4).

Location is not content. A single layer carries much of the donor's benefit, yet scaling, contrastive, fitted, and averaged transformations at the same layer do not reproduce it (Tab. 4 and Appendix H). The tested readouts recover the hint type and the predicted convention more reliably than the resulting intent decision (Sec. 5.6), which suggests that an effective intervention needs to supply decision-specific information rather than amplify a fixed direction, although the readouts and transformations we tested do not exclude other forms.

Limitations. The environment is a Hanabi-derived two-player game with fixed sender-specific conventions and scripted hint generation, so the results concern controlled receiving decisions rather than autonomous cooperation, and games with more than two players are left to future work. Within this environment, the probes are trained on scripted interactions, so their transfer to model-generated histories depends on how the available evidence changes, and the donor-transfer gains are measured on recorded states and do not establish gains over complete games. The mechanistic claims are narrower still, because the activation analyses cover only Qwen3-8B and the readouts and fixed transformations are linear functions of attention-block outputs at the answer position, so information in the residual stream or the MLP outputs, at earlier positions, or in nonlinear form is not excluded.