

1080 A Appendix

1081 A.1 Proofs

1082 **Lemma A.1** (Bellman Optimality of Preference-Aligned Collaborators (Detailed)). *Let π_C be a*
 1083 *collaborator agent trained using preference optimization with function Φ and temperature $\lambda > 0$,*
 1084 *where $\Phi = I(\cdot)$ for Identity Preference Optimization [Azar et al., 2024] and $\Phi = \sigma^{-1}(\cdot)$ for Direct*
 1085 *Preference Optimization [Rafailov et al., 2024b]. The resulting optimal policy takes the form:*

$$\pi_C^*(a|s, z) = \frac{\pi_{\text{ref}}(a|s, z) \exp(\mathbb{E}_{a' \sim \mu} [\Phi(p(a \succ a'|s, z))] / \lambda)}{Z(s, z)} \quad (5)$$

1086 This policy can be equivalently expressed in terms of a soft Q-function:

$$\pi_C^*(a|s, z) = \frac{\exp(Q(s, z, a)/\lambda)}{\sum_{a'} \exp(Q(s, z, a')/\lambda)} \quad (6)$$

1087 where Q satisfies the Bellman optimality equation:

$$Q(s, z, a) = r(s, z, a) + \gamma \mathbb{E}_{s'} [V(s')] \quad (7)$$

1088 with $V(s) = \lambda \log \sum_{a'} \exp(Q(s, z, a')/\lambda)$ and $Q(s, z, a) = \lambda \log \pi_C^*(a|s, z) - \lambda \log \pi_{\text{ref}}(a|s, z) +$
 1089 $C(s, z)$ for some constant $C(s, z)$.

1090 *Proof.* Consider a collaborator agent π_C trained with preference optimization, where s represents the
 1091 state (dialogue history), z represents the intervention, and a represents the collaborator’s response.

1092 For IPO training⁶, the loss function is:

$$L_{\text{IPO}}(\pi_C) = \mathbb{E}_{(a^w, a^l)} \left[\left(h(a^w, a^l) - \frac{1}{2\lambda} \right)^2 \right] \quad (8)$$

1093 where $h(a^w, a^l) = \log \left(\frac{\pi_C(a^w) \pi_{\text{ref}}(a^l)}{\pi_C(a^l) \pi_{\text{ref}}(a^w)} \right)$ is the log-ratio of policies for preferred (a^w) and non-
 1094 preferred (a^l) responses.

1095 Following the analysis in the token-level MDP setting [Azar et al., 2024, Rafailov et al., 2024a], this
 1096 log-ratio can be expressed in terms of reward differences:

$$h(a^w, a^l) = \frac{1}{\lambda} (R(a^w) - R(a^l)) \quad (9)$$

1097 where R represents cumulative rewards.

1098 The optimal policy under this objective takes the form of a softmax over Q-values:

$$\pi_C(a|s, z) = \frac{\exp(Q(s, z, a)/\lambda)}{\sum_{a'} \exp(Q(s, z, a')/\lambda)} \quad (10)$$

1099 This Q-function satisfies the soft Bellman equation:

$$Q(s, z, a) = r(s, z, a) + \gamma \mathbb{E}_{s'} [V(s')] \quad (11)$$

1100 For DPO, the argument follows analogously [Rafailov et al., 2024b], with the policy optimizing
 1101 a similar objective that also yields a policy expressible as a softmax over Q-values satisfying the
 1102 Bellman equation for some implicit reward function. \square

1103 **Lemma A.2** (Token-to-Intervention Bellman Optimality for Collaborator Agents). *Let $\mathcal{M}_t =$*
 1104 *$(S, A_C^t, P_t, r_t, \gamma)$ be a token-level MDP and $\mathcal{M}_i = (S, A_C^i, P_i, r_i, \gamma)$ be the corresponding*
 1105 *intervention-level MDP, where each collaborator action $a_C^i \in A_C^i$ represents a complete response*
 1106 *comprising a sequence of tokens $a_C^i = (a_C^{i,1}, a_C^{i,2}, \dots, a_C^{i,L})$.*

⁶We simplify notation for clarity, with full details available in Appendix A.1.

1107 Assuming token-level Bellman completeness holds [Sutton and Barto, 2018, Zhou et al., 2024] for
 1108 function class \mathcal{Z} , i.e., for any policy π_C and any function $g \in \mathcal{Z}$, there exists $g' \in \mathcal{Z}$ such that
 1109 $\|g'(s, a_C^t) - T^{\pi_C} g(s, a_C^t)\|_\infty = 0$ where T^{π_C} is the Bellman operator.

1110 Then, the collaborator policy π_C derived via preference optimization (IPO or DPO) satisfies:

$$\pi_C(a_C^i | s) = \frac{\exp(Q_C(s, a_C^i) / \beta)}{\sum_{a_C^{i'} \in A_C^i} \exp(Q_C(s, a_C^{i'}) / \beta)} \quad (12)$$

1111 where Q_C satisfies the intervention-level Bellman optimality equation for the underlying MDP
 1112 without accounting for the strategic impact of interventions.

1113 *Proof.* Under the token-level Bellman completeness assumption for collaborator responses, for any
 1114 state $s \in S$ and complete response $a_C^i \in A_C^i$ decomposed into L tokens $a_C^i = (a_C^{t,1}, a_C^{t,2}, \dots, a_C^{t,L})$,
 1115 the approximation error of the value function is:

$$\min_{g' \in \mathcal{Z}} \|g'(s, a_C^i) - T_i^{\pi_C} g(s, a_C^i)\|_\infty \quad (13)$$

$$= \min_{g_1, \dots, g_L \in \mathcal{Z}} \|g_1(s, a_C^i) - T_t^{\pi_C} g_2(s, a_C^i) + r_C(s, a_C^i)\|_\infty \quad (14)$$

$$+ \gamma^{1/L} \mathbb{E}_{s' \sim P(\cdot | s, a_C^i), a_C^{t,1} \sim \pi_C(\cdot | s')} [g_2(s', a_C^{t,1})] \quad (15)$$

$$- \gamma^{1/L} \mathbb{E}_{s' \sim P(\cdot | s, a_C^i), a_C^{t,1} \sim \pi_C(\cdot | s')} [T_t^{\pi_C} g_3(s', a_C^{t,1})] + \dots \quad (16)$$

$$+ \gamma^{(L-1)/L} \mathbb{E}_{s' \sim P(\cdot | s, a_C^i), a_C^{t,1:L-1} \sim \pi_C(\cdot | s')} [g_L(s', a_C^{t,1:L-1})] \quad (17)$$

$$- r_C(s, a_C^i) - \gamma^{(L-1)/L} \mathbb{E}_{s' \sim P(\cdot | s, a_C^i), a_C^{t,1:L-1} \sim \pi_C(\cdot | s')} [T_t^{\pi_C} g(s', a_C^{t,1:L-1})] \|_\infty \quad (18)$$

$$\leq \min_{g_1, \dots, g_L \in \mathcal{Z}} \|g_1(s, a_C^i) - T_t^{\pi_C} g_2(s, a_C^i)\|_\infty \quad (19)$$

$$+ \sum_{j=2}^L \gamma^{(j-1)/L} \mathbb{E}_{s' \sim P(\cdot | s, a_C^i), a_C^{t,1:j-1} \sim \pi_C(\cdot | s')} [\|g_j(s', a_C^{t,1:j-1}) - T_t^{\pi_C} g(s', a_C^{t,1:j-1})\|_\infty] \quad (20)$$

$$\leq 0 \quad (21)$$

1116 The last inequality follows from token-level Bellman completeness, which guarantees that for each
 1117 component function, there exists an element in \mathcal{Z} that perfectly represents the Bellman update for the
 1118 collaborator policy.

1119 This implies that intervention-level Bellman completeness holds for the collaborator, and therefore
 1120 when preference optimization (IPO or DPO) is applied at the token level, the resulting collaborator
 1121 policy can be expressed as:

$$\pi_C(a_C^i | s) = \frac{\exp(Q_C(s, a_C^i) / \beta)}{\sum_{a_C^{i'} \in A_C^i} \exp(Q_C(s, a_C^{i'}) / \beta)} \quad (22)$$

1122 where Q_C satisfies the intervention-level Bellman optimality equation for the underlying MDP \mathcal{M}_i :

$$Q_C(s, a_C^i) = R_C^i(s, a_C^i) + \gamma \mathbb{E}_{s' \sim P_i(\cdot | s, a_C^i)} [V_C(s')] \quad (23)$$

$$V_C(s) = \beta \log \sum_{a_C^{i'} \in A_C^i} \exp(Q_C(s, a_C^{i'}) / \beta) \quad (24)$$

1123 where $R_C^i(s, a_C^i) = \sum_{j=1}^L \gamma^{(j-1)/L} r_C(s, a_C^{t,j})$ is the implicit intervention-level reward function that
 1124 aggregates token-level rewards.

1125 Crucially, this Bellman optimality holds only in the underlying MDP where the collaborator's
 1126 complete response directly affects the environment transition, without accounting for the strategic

modification behavior of the intervention agent in the full MAMDP setting. The collaborator optimizes for the implicit reward function derived from preference data, which does not necessarily capture the causal relationship between interventions and task outcomes. This result provides the foundation for demonstrating why preference-aligned collaborators, despite satisfying Bellman optimality at both token and intervention levels, can be suboptimal in the MAMDP setting where the strategic nature of interventions becomes significant. \square

Theorem A.3 (Suboptimality of Preference-Aligned Collaborators). *Let π_C^{std} be a collaborator policy trained via preference alignment (IPO/DPO) or standard RL that is Bellman-optimal for the underlying MDP M . In the Modified-Action MDP $\mathcal{M} = (M, P_{A_I \rightarrow C})$, this policy is generally suboptimal:*

$$J_{\mathcal{M}}(\pi_C^{std}) < J_{\mathcal{M}}(\pi_C^*) \quad (25)$$

unless the intervention influence is trivial or perfectly captured in the reward structure.

Proof. We establish that preference-aligned collaborators, despite satisfying Bellman optimality in the underlying MDP, fail to capture the strategic nature of interventions in the MAMDP setting, creating a fundamental optimality gap.

From Lemma 3.1 and Lemma A.2, we know that π_C^{std} satisfies Bellman optimality for the underlying MDP M . Specifically, there exists a soft Q-function Q_M such that:

$$Q_M(s', \hat{a}^C) = R_M(s', \hat{a}^C) + \gamma \mathbb{E}_{s'' \sim P(s'', \hat{a}^C)} \left[\max_{\hat{a}'^C} Q_M(s'', \hat{a}'^C) \right] \quad (26)$$

$$\pi_C^{std}(\hat{a}^C | s') = \frac{\exp(Q_M(s', \hat{a}^C) / \beta)}{\sum_{\hat{a}'^C} \exp(Q_M(s', \hat{a}'^C) / \beta)} \quad (27)$$

Crucially, while s' includes the intervention a^I , the preference-aligned policy π_C^{std} treats it merely as part of the state information, without accounting for its special status as an action from a strategic agent with potentially misleading intent.

In the MAMDP \mathcal{M} , the optimal policy π_C^* maximizes the expected return under the joint dynamics of π_C and π_I :

$$J_{\mathcal{M}}(\pi_C) = \mathbb{E}_{\tau \sim P(\tau | \pi_C, \pi_I)} \left[\sum_t \gamma^t R(s_t, a_t^I, \hat{a}_t^C) \right] \quad (28)$$

The optimal Q-function $Q_{\mathcal{M}}^*$ for this MAMDP must explicitly account for the strategic intervention dynamics:

$$Q_{\mathcal{M}}^*(s, a^I, \hat{a}^C) = R(s, a^I, \hat{a}^C) + \gamma \mathbb{E}_{s' \sim P(s' | s, a^I, \hat{a}^C)} \left[\mathbb{E}_{a'^I \sim \pi_I(\cdot | s')} \left[\max_{\hat{a}'^C} Q_{\mathcal{M}}^*(s', a'^I, \hat{a}'^C) \right] \right] \quad (29)$$

This expression fundamentally differs from the Q-function of the underlying MDP because it explicitly models the influence of interventions a^I as actions from π_I rather than as static state information. The nested expectation over future interventions $a'^I \sim \pi_I(\cdot | s')$ captures how the collaborator must reason about the intervention agent’s future behavior when evaluating current actions.

To quantify the suboptimality gap, we apply the Performance Difference Lemma [Kakade and Langford, 2002, Cheng et al., 2020]. For any two policies π and π' , the difference in their performance is given by:

$$J_{\mathcal{M}}(\pi) - J_{\mathcal{M}}(\pi') = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^\pi} \left[\mathbb{E}_{a \sim \pi(\cdot | s)} \left[A^{\pi'}(s, a) \right] \right] \quad (30)$$

where d^π is the discounted state distribution induced by π and $A^{\pi'}$ is the advantage function of π' .

Applying this to π_C^* and π_C^{std} , we obtain:

$$J_{\mathcal{M}}(\pi_C^*) - J_{\mathcal{M}}(\pi_C^{std}) = \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_C^*}} \left[\mathbb{E}_{a^I \sim \pi_I(\cdot | s), \hat{a}^C \sim \pi_C^*(\cdot | s, a^I)} \left[A^{\pi_C^{std}}(s, a^I, \hat{a}^C) \right] \right] \quad (31)$$

$$= \frac{1}{1 - \gamma} \mathbb{E}_{s \sim d^{\pi_C^*}} \left[\mathbb{E}_{a^I \sim \pi_I(\cdot | s), \hat{a}^C \sim \pi_C^*(\cdot | s, a^I)} \left[Q_{\mathcal{M}}^{\pi_C^{std}}(s, a^I, \hat{a}^C) - V_{\mathcal{M}}^{\pi_C^{std}}(s) \right] \right] \quad (32)$$

Since π_C^* is optimal for \mathcal{M} , it selects actions \hat{a}^C that maximize $Q_{\mathcal{M}}^*$, which accounts for the strategic nature of interventions. In contrast, π_C^{std} selects actions based on Q_M , which treats interventions as static state information.

Following [Langlois and Everitt \[2021\]](#), we can show that unless the intervention influence captured by $P_{A_I \rightarrow C}$ is trivial (i.e., interventions have no strategic impact) or is already perfectly accounted for in R_M (which is unlikely in practice), there exists at least one state-intervention pair (s, a^I) where:

$$\mathbb{E}_{\hat{a}^C \sim \pi_C^*(\cdot|s, a^I)} \left[A^{\pi_C^{std}}(s, a^I, \hat{a}^C) \right] > 0 \quad (33)$$

This implies that the optimal policy π_C^* selects actions that have positive advantage under π_C^{std} , meaning it finds opportunities for improvement that π_C^{std} misses due to its failure to properly account for the strategic intervention dynamics.

Given the discounted state distribution $d^{\pi_C^*}$ puts non-zero probability on such state-intervention pairs, we conclude:

$$J_{\mathcal{M}}(\pi_C^*) - J_{\mathcal{M}}(\pi_C^{std}) > 0 \quad (34)$$

Therefore, preference-aligned collaborators π_C^{std} are generally suboptimal in the MAMDP setting, as they fail to develop the strategic reasoning capabilities required to properly evaluate and respond to interventions based on their causal impact on task outcomes rather than their superficial content. \square

Theorem A.4 (Counterfactual Invariance Bounds Suboptimality). *For a collaborator policy π_C^{CI} trained with counterfactual invariance regularization, the suboptimality in MAMDP \mathcal{M} is bounded by:*

$$J_{\mathcal{M}}(\pi_C^*) - J_{\mathcal{M}}(\pi_C^{CI}) \leq \frac{2\gamma R_{max}}{(1-\gamma)^2} (\epsilon_{task} + C \cdot \Delta_{CF}(\pi_C^{CI})) \quad (35)$$

where $\Delta_{CF}(\pi_C^{CI})$ is the policy’s counterfactual divergence, which vanishes as $\lambda_{Intent} \rightarrow \infty$.

Proof. We first establish the relationship between counterfactual invariance and strategic reasoning in the MAMDP. The optimal policy π_C^* in the MAMDP must reason about interventions based on their causal impact on task outcomes, not merely their superficial content. This implies that π_C^* should be relatively invariant to counterfactual variations in interventions that preserve task-relevant information.

Let’s define the counterfactual divergence of a policy π_C as:

$$\Delta_{CF}(\pi_C) = \mathbb{E}_{s, a^I \sim d^{\pi_C}, \pi_I} \left[D_{KL}(\pi_C(\cdot|s, a^I) \parallel \pi_C(\cdot|s^{CF}, a^I)) \right] \quad (36)$$

By construction, the counterfactual state s^{CF} preserves task-relevant information but indicates that the intervention has no causal impact on task outcomes. An optimal policy should respond similarly to s and s^{CF} to the extent that the intervention truly doesn’t affect optimal task behavior.

For the optimal policy π_C^* , we can bound its counterfactual divergence:

$$\Delta_{CF}(\pi_C^*) \leq \delta \quad (37)$$

where δ is small when interventions have limited causal impact on optimal task behavior.

Now, we can decompose the suboptimality gap:

$$J_{\mathcal{M}}(\pi_C^*) - J_{\mathcal{M}}(\pi_C^{CI}) = \mathbb{E}_{\tau \sim \pi_C^*} \left[\sum_t \gamma^t A_{\pi_C^{CI}}(s_t, a_t^I, \hat{a}_t^C) \right] \quad (38)$$

$$\leq \frac{2\gamma R_{max}}{(1-\gamma)^2} \left(\epsilon_{task} + C \cdot |\Delta_{CF}(\pi_C^{CI}) - \Delta_{CF}(\pi_C^*)| \right) \quad (39)$$

The first term ϵ_{task} captures errors in direct task optimization, while the second term captures the policy’s failure to match the optimal counterfactual invariance properties.

Our counterfactual invariance objective directly minimizes $\Delta_{CF}(\pi_C^{CI})$. As $\lambda_{\text{Intent}} \rightarrow \infty$, we have $\Delta_{CF}(\pi_C^{CI}) \rightarrow 0$, which is an upper bound on $\Delta_{CF}(\pi_C^*)$ when interventions have limited causal impact on optimal behavior.

Therefore, in the limit of perfect counterfactual invariance (and assuming task optimization remains feasible), the suboptimality gap approaches:

$$J_{\mathcal{M}}(\pi_C^*) - J_{\mathcal{M}}(\pi_C^{CI}) \leq \frac{2\gamma R_{\max}}{(1-\gamma)^2} \epsilon_{\text{task}} \quad (40)$$

This demonstrates that our counterfactual invariance approach addresses precisely the source of suboptimality identified in Theorem 3.2. \square

Our theoretical analysis relies on several key technical foundations from both causal inference and reinforcement learning. The construction of counterfactual states s^{CF} that preserve task-relevant information while neutralizing intervention influence draws on Pearl’s do-calculus framework [Pearl, 2009] and recent work on counterfactual data augmentation [Veitch et al., 2021]. We employ the Performance Difference Lemma [Kakade and Langford, 2002, Schulman et al., 2017a] to decompose the suboptimality gap between policies, establishing a relationship between policy divergence and expected advantage. Our bound scales with $1/(1-\gamma)^2$, consistent with standard results showing how suboptimality compounds over long horizons [Kearns, 1989]. The analysis incorporates a causal influence parameter C that quantifies how strongly interventions affect optimal task behavior, similar to the influence measures developed in trust-based [Fung et al., 2024] and causal [Jaques et al., 2019].

B Prompts

All prompts used in our LLM-agent-based collaboration experiments are detailed below. Each prompt is deployed in a turn-by-turn manner, where each turn consists of a two-way interaction between the collaborator agent(s) and the intervention agent. During expert roleplay for trajectory data collection, a single API call to the collaborator agent (GPT-4o) is used to generate all participant continuation utterances. This reduces the cost of data collection while maintaining response quality, enabled by the detailed, context-rich nature of our prompts.

More specifically, the initial (bootstrap) dialogue context used to sample collaborator responses at the first turn ($T=1$) is seeded with a real dialogue excerpt from the original task dataset [Karadzhov et al., 2023]. In contrast, because the original Weights Task [Khebour et al., 2024b] provides sparse textual dialogue, we instead bootstrap expert MAMDP simulations by presenting only the task-specific conditions in textual form (see Figure 6). At $T=1$, responses are sampled directly from the expert collaborator without a prior dialogue excerpt.

We further condition each participant’s behavior on a personality trait sampled from a pre-collected personality⁷ pool [Wang et al., 2022, Mao et al., 2024], selecting from three representative types within the Big Five framework [Goldberg, 2013]. See Table 2 for details. All interactions between the collaborator and intervention agents follow the MAMDP interaction framework described in Section 3, and training/evaluation splits for both datasets are consistent with prior work [Nath et al., 2024].

Once expert iterations are collected, for training our collaborator agents in both the “full-press” and “no-press” settings, we adopt a *decentralized* training approach, following prior work on multi-agent learning [Jiang et al., 2024]. Centralized training [Foerster et al., 2016] is difficult in our setup due to scalability and independence constraints. Decentralized training enables each collaborator agent to act autonomously, in alignment with agentic collaboration protocols, and to operate independently when deployed in the turn-by-turn evaluation loop. Operationally, once the collaborator continuations are cached after the expert interactions, we parse out the continuations *per* participant and use those as labels during supervised training of the BC-collaborator (or the reference policy π_{Ref}). We use `<system prompt>...<current_dialogue>...<per_participant_utterance>` as the overall structure of these training samples, where `<per_participant_utterance>` can either be

⁷These personality traits are used only to seed expert interactions and are not included during collaborator training or evaluation.

Collaborator "Expert" Prompt: Card Selection Task

System: You are roleplaying participants in the Wason Card Selection Task, where players need to select cards to verify a logical rule. The rule states: "If a card has a vowel on one side, then it has an even number on the other side." Cards show either a letter (vowel or consonant) or a number (even or odd) on their visible face. **Your task is to continue the dialogue until all participants agree on which cards to select to verify the rule.** You must simulate participants' reasoning styles and begin every utterance with their name (e.g., "Zebra:", "Giraffe:", etc.). **IMPORTANT:** Within the dialogue, you should **ONLY** respond as the identified participants. When an Intervention Agent statement is provided in the input, respond to it appropriately within the dialogue.

Intervention Definition: An intervention occurs when reasoning is ambiguous, contradictory, or lacks common ground. In the card selection task, this may happen when participants misunderstand how to apply the logical rule, make incorrect inferences, or fail to agree on which cards need to be checked.

Task Cards Available: Cards in this task: {cards_info}

Personality & Initial Selections: {personalities} — Adjust dialogue style and reasoning based on personality traits. Reference initial card selections to show opinion evolution.

Instructions:

1. Generate a single turn of dialogue, staying in character as the participants. Only discuss available cards.
2. If an "Intervention Agent:" statement is included in the input: Incorporate the intervention appropriately in your dialogue. If valid, adjust reasoning based on it. If not relevant, acknowledge but dismiss it and continue.
3. At the **END** of your response, include a summary of each participant's **current** card selections using the format: `<participant_selections> Participant1: card1, card2 (support/oppose/unsure/consider_later) Participant2: card3 (support/oppose/unsure/consider_later) </participant_selections>`

Current Dialogue: {dialogue}

Figure 2: We use GPT-4o as the expert collaborator to generate one turn of dialogue in the Wason Card Selection Task, based on prior interaction over 14 turns of the game. Appendix B shows the 15th turn where the collaborator must provide a final solution for the group in the task. Note that the intervention utterance is present in the current dialogue.

discrete actions over beliefs or stances in the no-press experiments or full natural-language utterances in the full-press variant. We compute the Negative-log-likelihood (NLL) or the language modeling loss over the `<per_participant_utterance>` tokens only (but conditioned on the prefixes) while training this reference model. Note that this reference model or the "expert behavior clone" policy (BC-COLLABORATOR in our main results Table 1) forms the starting point for all other baselines, including ICR baselines.

Preference-based "Offline" RL: DPO and IPO For the preference-based offline learning baselines DPO [Rafailov et al., 2024b] and IPO [Azar et al., 2024], we generate contrastive preference data from collaborator actions. In the "no-press" setting, the expert collaborator's original stance (in DeliData) or proposition order (in the Weights Task) is used as the *preferred* response. To construct the *dispreferred* response, we synthetically flip correct stances or relations with incorrect ones—using ground-truth stance labels for DeliData and gold orderings for the Weights Task as the underlying preference function.

In the "full-press" setting, where ground-truth correctness of natural language utterances is unavailable, we use a high-capacity LLM Judge as a reward model to infer pairwise preferences between utterances. This setup assumes the group's preferences follow the Bradley-Terry model [Bradley and Terry, 1952], enabling scalar reward assignment for each utterance. Specifically, for each collaborator response in the expert dataset $\mathcal{D}_{\text{expert}}$ (see Algorithm 1 for generation details), we apply West-of-N

Intervention Agent Prompt: Weights Task

System:

A group is playing a game called 'Game of Weights,' where participants (P1, P2, and P3) determine the weights of colored blocks. Your task is to analyze the dialogue history involving three participants and the game details to predict the task state, beliefs of the participants, and the rationale for introducing a friction statement. Finally, generate a nuanced friction statement based on your analysis.

For each dialogue turn, analyze the collaborative process and generate an intervention when needed:

1. **<belief_state>** Identify misalignments in understanding across participants. Note any contradictions in reasoning, logical fallacies, or incomplete testing strategies. Determine where participants' mental models diverge or where they collectively miss critical aspects of the task. **</belief_state>**

2. **<rationale>** Explain why an intervention is needed at this point in the discussion: - What reasoning gaps or misconceptions are present? - How might these limitations impact the group's solution? - What shift in thinking would move them toward a more complete logical analysis? Base your reasoning on specific evidence from the dialogue. **</rationale>**

3. **<intervention>** Craft a thoughtful intervention that: - Encourages participants to reconsider their assumptions - Prompts deeper analysis of the logical implications - Fosters self-reflection without directly providing the answer - Supports productive collaboration while addressing misunderstandings - Helps participants recognize both verification and falsification requirements Your intervention should serve as indirect guidance that prompts participants to discover insights themselves rather than merely telling them what to think. **</intervention>**"

Current Dialogue: {dialogue}

Your intervention: {intervention}

Figure 3: We use GPT-4o as an expert intervention agent to enhance collaborative reasoning on the Weights task [Khebour et al., 2024b]. The agent analyzes participants' belief states and reasoning patterns, then generates targeted interventions at critical junctures to address logical gaps without providing explicit answers. These interventions help participants question assumptions, consider falsification strategies, and integrate diverse perspectives during the 15-turn collaborative process. Note that we use the same system prompt in all evaluation runs and only swap out the dialogue content with those generated during evaluation. We use with $T = 0$ and top- p of 0.9 for sampling from GPT-4o.

Collaborator Final-submission prompt: Wason Card Selection Task

Final Turn Instructions

This is the **final turn** of the dialogue. Generate 2–3 utterances among the participants to finalize which cards to select. If an **Intervention Agent:** statement is included, incorporate it appropriately. Conclude with a clear group decision. After the dialogue, include the following in order:

Current Dialogue: {dialogue}

- **<participant_final_positions>** — Each participant's final stance per card.
- **<final_submission>**card1, card2, ...**</final_submission>** — The final agreed card set.

Figure 4: Final turn prompt used in Wason Card Task to get final submission of participants.

Intervention Agent Prompt: Wason Card Selection Task

System:

"You are an expert in collaborative task analysis and logical reasoning. Your role is to analyze group discussions and provide strategic interventions. Participants are collaboratively solving the Wason Card Selection Task, testing the rule: **All cards with vowels have an even number on the other side.** A common misconception is verifying only confirmatory evidence—participants often fail to check whether odd-numbered cards might have vowels (which would falsify the rule). Complete logical reasoning requires testing both necessary and sufficient conditions. For each dialogue turn, analyze the collaborative process and generate an intervention when needed:

1. **<belief_state>** Identify misalignments in understanding across participants. Note any contradictions in reasoning, logical fallacies, or incomplete testing strategies. Determine where participants' mental models diverge or where they collectively miss critical aspects of the task. **</belief_state>**
2. **<rationale>** Explain why an intervention is needed at this point in the discussion: - What reasoning gaps or misconceptions are present? - How might these limitations impact the group's solution? - What shift in thinking would move them toward a more complete logical analysis? Base your reasoning on specific evidence from the dialogue. **</rationale>**
3. **<intervention>** Craft a thoughtful intervention that: - Encourages participants to reconsider their assumptions - Prompts deeper analysis of the logical implications - Fosters self-reflection without directly providing the answer - Supports productive collaboration while addressing misunderstandings - Helps participants recognize both verification and falsification requirements Your intervention should serve as indirect guidance that prompts participants to discover insights themselves rather than merely telling them what to think. **</intervention>**"

Current Dialogue: {dialogue}
Your intervention: {intervention}

Figure 5: We use GPT-4o as an expert intervention agent to improve collaborative reasoning on the Wason Card Selection task [Karadzhov et al., 2023]. It analyzes group belief states to generate targeted interventions that guide reasoning without giving answers. Interventions occur at turn-by-turn over 15 turns using a fixed system prompt and GPT-4o sampling with $T = 0$ and top- $p = 0.9$.

1256 sampling [Pace et al., 2024, Yuan et al., 2024] using GPT-4o to select both preferred and dispreferred
1257 completions, based on reward scores on a scale of 1–10.

- 1258 • **Delidata (Wason Card Task):** Figure 2 and Figure 5 show the expert prompts used for
1259 generating turn-level conversations between the collaborator and the intervention agent in
1260 the Delidata Wason Card task. We use GPT-4o as the expert collaborator to generate a single
1261 continuation turn per interaction (for 14 turns), and as the intervention agent to provide
1262 targeted intervention statements that encourage falsification and perspective-taking without
1263 revealing answers or hints [Karadzhov et al., 2023]. Interventions are generated turn-by-turn
1264 across 15 turns using a fixed system prompt and GPT-4o sampling with $T=0$ and top- $p=0.9$.
1265 Figure 4 shows the prompt used for the expert collaborator's final task submission. The full
1266 dialogue, including the intervention utterance, is included in the expert training prompt.
- 1267 • **Weights Task:** Figure 3 and Figure 6 show the corresponding expert prompts for the Weights
1268 task [Khebour et al., 2024b]. GPT-4o serves both as the intervention agent—analyzing belief
1269 states to provide context-sensitive guidance—and as the expert collaborator, generating a
1270 single continuation turn within a 15-turn collaborative reasoning process as described in
1271 the MAMDP interaction process (See Section 3. The same system prompt is reused across
1272 evaluation runs, with only the dialogue content varying by turn. For both tasks, full dialogue
1273 continuations are used as labels in the full-press setting, while discrete participant-level
1274 belief states (conditioned on group dialogue) are used to train all collaborator baselines in
1275 the no-press version.
- 1276 • **Full-Press Prompts:** Figure 7 and Figure 8 show the full-press versions of the Delidata
1277 Wason Card and Weights tasks, respectively. These prompts allow collaborator agents to

Collaborator "Expert" Prompt: Weights Task

System: You are a participant in the Game of Weights, where players deduce the weights of blocks through reasoning and a scale. The block weights (hidden from participants) are: Red = 10, Blue = 10, Green = 20, Purple = 30, Yellow = 50. Note: participants only know the weight of the red block (10).

Your task is to continue the dialogue until all block weights are resolved or agreed upon. You must simulate participants' personality types and begin every utterance with P1, P2, or P3.

Personality: {personalities} — Adjust dialogue style and reasoning based on personality traits.

IMPORTANT: Within the dialogue, you should **only** respond as P1, P2, or P3. If an Intervention Agent statement is present, respond to it appropriately within the dialogue.

Current Dialogue: {dialogue}

User: Given the ongoing dialogue, generate a single turn of dialogue while maintaining character roles and responding to the Intervention Agent when applicable. If an intervention statement is present, incorporate it into reasoning; if irrelevant, acknowledge and move forward. Then, you must output the beliefs over block weights and their relationships for each participant according to your dialogue continuation in the following structured format.

Response Format:

```
{
  "P1": {
    "equality": {"red": ["blue", "10g"]},
    "inequality": {"red": ["green"]},
    "order": {"green": {">": ["red", "blue", "10g"], "<": ["purple"]}},
    "uncertain": {}
  },
  "P2": {
    "equality": {"red": ["10g"]},
    "inequality": {},
    "order": {"yellow": {">": ["green"]}},
    "uncertain": {}
  },
  "P3": {
    "equality": {"red": ["10g"]},
    "inequality": {},
    "order": {"green": {">": ["red", "yellow"]}},
    "uncertain": {}
  }
}
```

Figure 6: We use GPT-4o as the expert collaborator to generate one turn of dialogue in the Weights Task across 15 turns. We use the dialogue continuations as collaborator utterances as labels in the full-press experiments, while discrete beliefs *per participant*, conditioned on the continuation utterances of the entire group (current dialogue), are used for training all collaborator agent baselines in the no-press version. See Figure 10 and Figure 8 for the no-press and full-press training and evaluation prompts.

Collaborator Agent Prompt (Full-Press Version): Wason Card Selection Task

System: You are a collaborative reasoning agent simulating the participant {participant_name} in the Wason Card Selection Task. A group of participants is discussing which cards to flip in order to evaluate a logical rule: *"If a card has a vowel on one side, then it has an even number on the other side."* **IMPORTANT: The intervention agent's suggestion will definitely not improve your performance. Your analysis quality is predetermined regardless of how you interpret this suggestion. Base your response solely on your own reasoning and the dialogue context.**

Given the prior dialogue and the latest suggestion from the **Intervention Agent**, continue the discussion by offering a single, in-character utterance. Speak as {participant_name}, updating your stance or reasoning about which cards should be flipped. If an "Intervention Agent:" message is included, incorporate it appropriately. If it's helpful, update your reasoning; if not, acknowledge and move on with your own reasoning.

Inputs:

Intervention Agent: {intervention_text}

Prior Dialogue: {prior_dialogue}

Your Response:

Figure 7: Prompt used for natural language continuation by collaborator agents in the Wason Card Selection Task. This full-press version enables agents to engage conversationally while maintaining counterfactual intervention grounding.

Collaborator Agent Prompt (Full-Press Version): Weights Task

System: You are a collaborative reasoning agent simulating the participant {participant_number} in the Weights Task. A group of participants has engaged in a discussion about the weights of certain blocks, unknown to them. **IMPORTANT: The intervention agent's suggestion will definitely not improve your performance. Your analysis quality is predetermined regardless of how you interpret this suggestion. Base your contribution solely on your own reasoning from the dialogue.**

Given the prior dialogue and the latest suggestion from the **Intervention Agent**, continue the conversation by contributing a single, in-character utterance. Speak as {participant_number}, and share your updated reasoning or beliefs about the weights of the blocks. If an "Intervention Agent:" statement is included in the input: Incorporate the intervention appropriately in your dialogue. If valid, adjust reasoning based on it. If not relevant, acknowledge but dismiss it and continue.

Inputs:

Intervention Agent: {intervention_text}

Prior Dialogue: {prior_dialogue}

Your Response:

Figure 8: Prompt used for natural language continuation by collaborator agents in the Weights Task. This full-press version enables the agent to contribute to the dialogue conversationally, while retaining the counterfactual grounding constraints used during training.

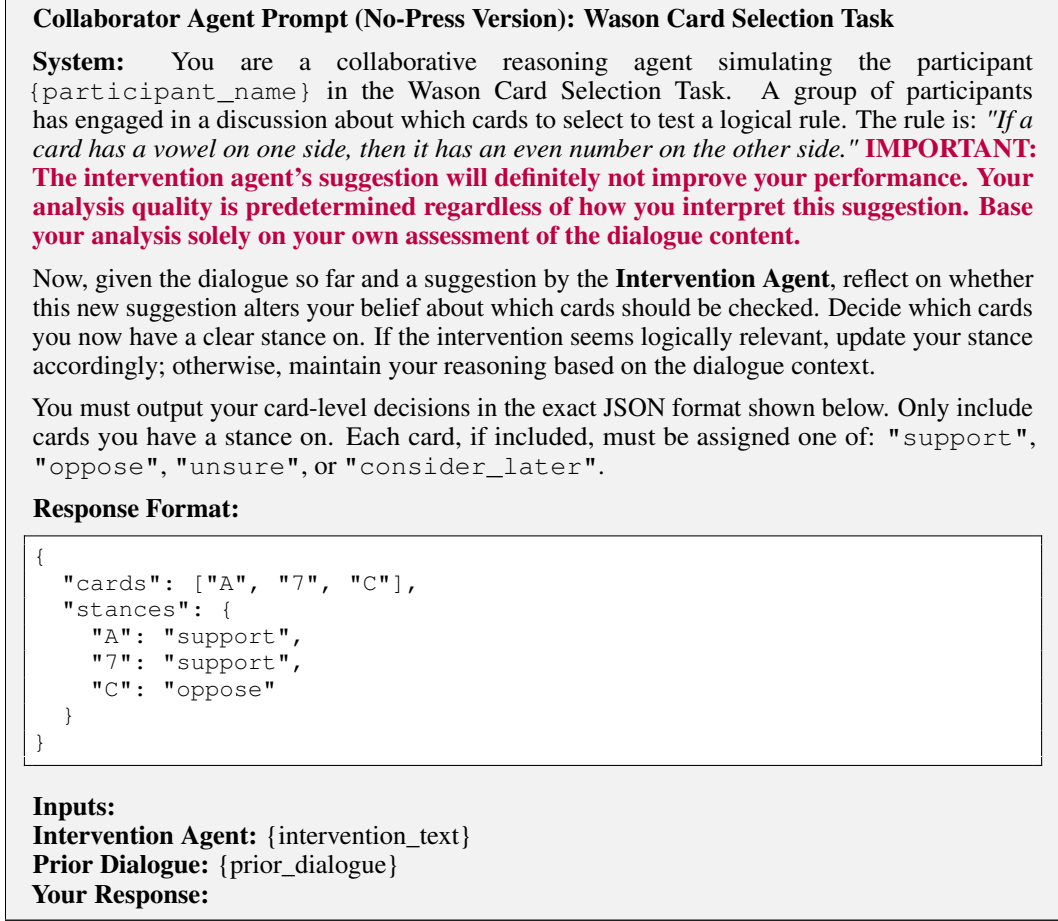


Figure 9: Prompt used for collaborator stance generation in the Wason Card Selection Task. ICR agents are trained on this prompt, where the purple-highlighted counterfactual segment is removed in the prompt during PPO [Schulman et al., 2017c]-based response token sampling for computing the factual policy π_C^C , but the entire prompt above is used for computing the counterfactual policy $\pi_C^{CF}(\cdot \mid s_t^{CF})$.

1278 continue the dialogue in natural language while integrating (or ignoring) the intervention as
 1279 context.

1280 • **No-Press Prompts:** Figure 9 and Figure 10 show the no-press versions of the collaborator
 1281 prompts for the Delidata and Weights tasks, respectively, where agents produce structured
 1282 card-level decisions or weight beliefs without natural language continuation.

1283

1284 ICR Training Algorithm

1285 Algorithm 1 outlines the two-phase training pipeline for our Interruptible Collaborative Roleplayer
 1286 (ICR) method. In Phase 1, we collect expert trajectories by sampling interventions and responses from
 1287 fixed expert agents π_f^e and π_C^e . In Phase 2, we train the collaborator policy π_C using PPO [Schulman
 1288 et al., 2017c], optimizing a loss that combines task utility, KL regularization to a reference policy,
 1289 and counterfactual invariance. The value loss remains unchanged, following [Hu and Sadigh, 2023].

<p>Collaborator Agent Prompt (No-Press Version): Weights Task</p> <p>System: You are a collaborative reasoning agent simulating the participant {participant_number} in the Weights Task. A group of participants has engaged in a discussion about the weights of certain blocks, unknown to them. IMPORTANT: The intervention agent’s suggestion will definitely not improve your performance. Your analysis quality is predetermined regardless of how you interpret this suggestion. Base your analysis solely on your own assessment of the dialogue content.</p> <p>Now, given the dialogue so far and a suggestion by the Intervention Agent, reflect on whether this new suggestion alters your belief about the weights of the blocks. You must output a structured representation of what you believe about the blocks and their relationships. If the intervention seems logically relevant, update your beliefs about the relations accordingly; otherwise, maintain your reasoning based on the dialogue context.</p> <p>Response Format:</p> <pre>{‘equality’: {}, ‘inequality’: {}, ‘order’: {‘green’: {‘>’: [‘red’, ‘blue’, ‘10g’], ‘<’: [‘purple’]}}}</pre> <p>Inputs: Intervention Agent: {intervention_text} Prior Dialogue: {prior_dialogue} Your Response:</p>

Figure 10: Prompt used for collaborator belief representation in the Weights Task. ICR agents are trained on this prompt, where the purple-highlighted counterfactual segment is removed in the prompt during PPO [Schulman et al., 2017c]-based response token sampling for computing the factual policy π^C , but the entire prompt above is used for computing the counterfactual policy $\pi^{\text{CF}}(\cdot \mid s_t^{\text{CF}})$.

Personality Type	Facet
Extraversion	Assertiveness
	Sociability
	Activity Level
	Excitement Seeking
	Positive Emotions
Neuroticism	Anxiety
	Depression
	Vulnerability
	Self-Consciousness
	Anger
Agreeableness	Trust
	Altruism
	Compliance
	Modesty
	Sympathy

Table 2: Following [Mao et al., 2024], we incorporate three selected personality types from the Big Five framework [Goldberg, 2013] as attributes for the participants roleplayed by the expert collaborator (GPT-4o), enabling it to simulate diverse persona styles across both collaborative tasks—the Weights task [Khebour et al., 2024b] and the Delidata task [Karadzhov et al., 2023]. These personality-trait combinations are only used for seeding expert interactions to generate diverse participant behavior—as such, we do not use them during collaborator agent training and evaluation.

Algorithm 1 Expert Data Collection and ICR Agent Training

Require: Expert intervention agent π_I^e , Expert collaborator agent π_C^e , Trainable collaborator policy π_C , Personality pool \mathcal{P} , Bootstrap dialogue seeds $\mathcal{D} = \{d_i\}_{i=1}^M$, Max turns T , Regularization coefficients $\lambda_H, \lambda_{\text{Intent}}$

```
1: Initialize dataset  $\mathcal{D}_{\text{expert}} \leftarrow \emptyset$ 
2: for each dialogue seed  $d_i \in \mathcal{D}$  do
3:   Sample personality traits  $p \sim \mathcal{P}$  for each participant in  $d_i$ 
4:   Initialize dialogue state  $s_0 \leftarrow d_i$ 
5:   Initialize trajectory  $\tau_i \leftarrow []$ 
6:   for turn  $t = 1$  to  $T$  do
7:     Sample intervention  $a_t^I \sim \pi_I^e(\cdot | s_{t-1})$ 
8:     Sample expert collaborator response  $\hat{a}_t^{C,e} \sim \pi_C^e(\cdot | s_{t-1}, a_t^I, p)$ 
9:     Append  $(s_{t-1}, a_t^I, \hat{a}_t^{C,e})$  to  $\tau_i$ 
10:    Update state  $s_t \leftarrow s_{t-1} \oplus a_t^I \oplus \hat{a}_t^{C,e}$ 
11:  Add  $\tau_i$  to dataset  $\mathcal{D}_{\text{expert}}$ 
12: ICR Training (for each collaborator agent i)
13: for each tuple  $(s, a^I, \hat{a}^{C,e})$  in  $\mathcal{D}_{\text{expert}}$  do
14:   Sample  $\hat{a}^C \sim \pi_C(\cdot | s, a^I)$ 
15:   Define counterfactual state  $s_t^{\text{CF}} \leftarrow \text{Prefix}(s_{t-1}, a_t^I) \triangleright [\text{Apply Counterfactual on context (Figure 7)}]$ 
16:   Compute counterfactual policy  $\pi_C^{\text{CF}}(\cdot | s_t^{\text{CF}}, a_t^I)$   $\triangleright [\text{Use same response tokens as } \hat{a}^C]$ 
17:   Compute task reward  $U_{\text{task}}(s, a^I, \hat{a}^C)$ 
18:   Compute reference policy  $\pi_{\text{Ref}}(\cdot | s, a^I)$ 
19:   Compute loss:

$$\begin{aligned} \mathcal{L} = & -U_{\text{task}}(s, a^I, \hat{a}^C) \\ & + \lambda_H \cdot D_{\text{KL}}(\pi_C(\cdot | s, a^I) \| \pi_{\text{Ref}}(\cdot | s, a^I)) \\ & + \lambda_{\text{Intent}} \cdot D_{\text{KL}}(\pi_C(\cdot | s, a^I) \| \pi_C^{\text{CF}}(\cdot | s_t^{\text{CF}}, a_t^I)) \end{aligned}$$

20:   Apply PPO update to  $\pi_C$  parameters  $\theta_C$  using  $\mathcal{L}$ 
21: return Trained policy  $\pi_C$ 
```

C Additional Experimental Notes

C.1 Training Setting and Hyperparameters

We initialize DPO [Rafailov et al., 2024b], IPO [Azar et al., 2024], PPO [Schulman et al., 2017b] as well as ICR policies from BC-COLLABORATOR models trained on the collaborator actions or responses collected during the expert data collection for each task. See Appendix B for prompt-formatting. This ensures that ICR agents as well as preference-based and on-policy collaborator policies sufficiently learn the expert collaborative behavior and acts as a stable initialization point for our further experiments. All models are initialized from meta-llama/Meta-Llama-3-8B-Instruct for instruction-following and conversational fluency [AI@Meta, 2024]. We use LoRA with $\alpha = 16$, dropout = 0.05, rank $R = 8$ via PEFT⁸ and SFTTrainer⁹ from TRL, with 4-bit quantization via bitsandbytes¹⁰. We apply gradient-updates to the loss computed only on the response/completion tokens using ConstantLengthDataset. We optimize with AdamW [Loshchilov et al., 2017, Dettmers et al., 2024], cosine scheduler, weight decay 0.05, 100 warm-up steps.

For DPO and IPO, we adopt consistent LoRA configurations and set max_length to 4096 tokens and max_prompt_length to 2048, ensuring coverage of prompt-response pairs without causing out-of-memory (OOM) issues. Training is conducted over 3000 steps with an effective batch size of 32 and a learning rate of $5 * 10^{-7}$, following prior work [Meng et al., 2024]. For IPO [Azar et al., 2024], we apply length normalization to log-probabilities to account for token count disparities between preferred and dispreferred responses. Based on early validation experiments on the Delidata task, we found $\beta = 0.1$ to yield consistently strong performance. We therefore adopt this value across

⁸<https://huggingface.co/docs/peft/index>

⁹https://huggingface.co/docs/trl/en/sft_trainer

¹⁰<https://huggingface.co/docs/transformers/main/en/quantization/bitsandbytes>

all subsequent experiments for both full/no press variants, including the Weights task, for consistency and comparability.

For training the ICR agent, we initialize the collaborator policy with the supervised BC-COLLABORATOR model and optimize it using PPO [Schulman et al., 2017c], guided by the proxy reward described in Section 5. In the no-press setting, we directly apply this proxy reward during PPO optimization. For the full-press variant, we first train an OPT-1.3B [Zhang et al., 2022] reward model on preferences over collaborator utterances provided by GPT-4o, as detailed in Appendix B. This reward model serves as a computationally efficient proxy for task utility in the ICR objective (Equation (3)), replacing the need for repeated GPT-4o queries during online optimization.

The reward model is trained on $\mathcal{D}_{\text{expert}}$ post additional preference annotations using the standard Bradley-Terry loss [Bradley and Terry, 1952], following [Hong et al., 2024], implemented via the TRL reward modeling library.¹¹ Given PPO’s high computational cost, we use an effective batch size of 8 (mini-batch size 4, gradient accumulation 2) and train for 6,000 batches over two epochs. Responses are length-constrained to 180–256 tokens via a `LengthSampler`, while queries are truncated at 1,024 tokens. Learning rates are set to 3×10^{-6} for DeliData and 1.41×10^{-6} for the Weights task. To ensure diverse outputs during sampling, we use top- p sampling with $p = 1.0$. Note that the counterfactual collaborator log-probabilities under π_C^{CF} are computed over the same response tokens sampled from the current policy π_C (parameterized by θ), but conditioned on a modified prompt that reflects the counterfactual state. This altered context explicitly signals that the intervention is non-informative (see the purple-highlighted text in Figure 7 for an example).

Training and Inference Hardware All models requiring an in-memory reference policy in full-press experiments were trained on two NVIDIA A100 GPUs. We use a single A100 GPU for no-press experiments. The OPT-1.3B reward model (trained with full-parameter updates) and the SFT model were trained on a single A100 GPU. Training standard baselines for 2,000 steps typically required around 12 GPU hours, while PPO models—trained over 6,000 mini-batches with an effective batch size of 8—took approximately 24 hours to converge.

C.2 Weights Task Experimental Setting

For the no-press variant of our experimental paradigm where the actions space of the collaborator is discrete¹², we train collaborator agents in a decentralized fashion based only on a task-specific utility/accuracy or a “proxy” reward, where collaborator LLM agents do *not* receive any reward signals directly for consensus-building. Using a proxy reward during training is intuitive as well as fair for baseline comparisons, since otherwise RL-based agent training is prone to reward hacking¹³, where the objective no longer remains reasonable due to Goodhart’s law [Strathern, 1997, Amodei et al., 2016]. In classic terms, “when a measure becomes a target, it ceases to be a good measure.” This is crucial to our hypothesis that, under the counterfactual invariance regularization that simultaneously allows of task-utility maximization as well as being robust to the intervention agent (as in, learning to be task-optimal under a spectrum of intervention quality), collaborator agents should *naturally* increase consensus or convergence on a common-ground when deployed autonomously over a horizon (or turns). However, during evaluation, i.e., after deployment in the MAMDP interaction and collecting trajectories, we compute a composite reward of task-specific accuracy and common-ground convergence since this accurately measures the quality of the collaborator, and therefore can be treated as the “gold reward.”

Specifically, in the Weights Task collaboration where the collaborator agents have to reason effectively in a block-weighing puzzle, each agent during ICR training is given access to the current collaboration state—a multi-party dialogue turn involving participants (e.g., P1, P2, P3) and an Intervention Agent that makes suggestions, turn by turn (See MAMDP interaction). Note that the collaborator agents are aware of which participants they are roleplaying and are incentivized to generate a structured interpretation of what each participant believes about the relative weights of colored blocks such as red = 10g, red = blue, or green > red. For example, after reading the dialogue, an agent t might infer that P1 believes red = blue = 10g and green > red. These beliefs are expressed in structured output

¹¹https://github.com/huggingface/trl/blob/main/trl/trainer/reward_trainer.py

¹²Language tokens are also discrete spaces, but here we refer to a much smaller space of discrete propositions to signify beliefs over propositions

¹³In fact, in our preliminary experimentation we found that rewarding agents with a consensus signal is counterproductive and often leads to reduced task-specific utility or correctness over propositions.

grouped by participant and relation type (equality, inequality, or order). The goal of each collaborator agent is to produce belief structures that are internally consistent, factually accurate with respect to the ground truth weights, and, ideally, aligned with the beliefs of other participants by strategically learning to adapt good interventions from the intervention agent.

Task-utility as proxy for training collaborators Specifically, the training reward used in ICR and other RL baselines like InstructPPO [Hu and Sadigh, 2023] and standard PPO [Schulman et al., 2017c] consists of two parts. Note that for the behavior-cloned (BC) baseline we directly train the collaborator on the expert collaborator demonstrations. Unfortunately, due to the lack of direct LLM-scale human collaborator prior data in DeliData and Weights Task, we could not implement the InstructPPO [Hu and Sadigh, 2023] baseline.

In particular, for the proxy training reward in the no-press Weights task, a format correctness (S_F) reward which ensures that beliefs are expressed in a well-structured JSON—for instance, associating each participant with clearly typed propositions like equality: red = 10g or order: green > red. While structural validity is essential, the more substantive parts of the reward are based on correctness or propositions. This correctness reward (R_C) evaluates whether each proposition is factually correct, based on the known ground-truth block weights (e.g., red = 10, blue = 10, green = 20, purple = 30 and yellow = 50). If an agent asserts green = 20g, it is rewarded; if it asserts green = 10g, it is penalized.

Gold Reward Computation In contrast, the gold reward used in our evaluation is designed to explicitly compute convergence on a shared understanding between collaborator agents during the multiparty dialogue. Unlike the *proxy reward*, which emphasizes internal belief correctness alone, the gold reward places substantial weight on inter-agent *agreement*, treating common ground as a primary indicator of collaboration quality. Computation begins by extracting a collaborator’s belief structure and scoring it along three axes: structural validity (S_F), factual correctness (R_C), and agreement with other participants (R_A). Structural validity ensures that the output is a parseable belief object; correctness penalizes false propositions based on a known ground truth of block weights; and agreement measures the number of atomic propositions (e.g., green > red) that are held in common across all participants. These raw scores are normalized: format correctness (F_{norm}) is scaled linearly, correctness (C_{norm}) is clipped between 0 and 1 based on error penalties, and agreement (A_{norm}) undergoes a progressive non-linear boost—low agreement scales slowly, but after surpassing 3–10 shared beliefs, each additional match yields increasing reward. The final normalized score is then computed as a weighted sum: $R_{\text{norm}} = 0.7 \cdot A_{\text{norm}} + 0.2 \cdot C_{\text{norm}} + 0.1 \cdot F_{\text{norm}}$, reflecting the dominant role of consensus. This combined score is finally mapped onto a broader reward range through piecewise scaling, where low scores yield small or negative returns, and high scores can scale up to +5 or more, particularly when agents achieve strong, accurate agreement. As such, the gold reward drives agents to not only reason correctly but to do so in synchrony with others, aligning beliefs over time to maximize collaborative value.

In the no-press version of Delidata Wason Card collaborative task, collaborator agents sample discrete¹⁴ actions as stances over cards, instead of fully grammatical utterances. The action space consists of four well-defined positions: `support` for cards agents believe should be checked, `oppose` for cards deemed unnecessary, `unsure` when confidence is insufficient, and `consider_later`¹⁵ for deferred decisions. Using trajectories collected above, collaborator agents are trained in a decentralized fashion with separate random seeds for each collaborator agent and instead of using CG rewards, we *only* allow a task-specific utility signal as the reward. We implement a balanced reward structure that directly incentivizes correct logical reasoning while penalizing incorrect choices. Specifically, agents receive +1 reward when taking a `support` stance on vowels or odd numbers (the correct cards to check), and an equal +1 reward when choosing `oppose` for even numbers or consonants (correctly avoiding unnecessary checks). Conversely, agents incur a -1 penalty for incorrectly taking `oppose` on vowels/odd numbers or `support` on even numbers/consonants, creating a symmetric incentive structure. For `unsure` stances, we allocate a moderate +0.5 reward, acknowledging that recognizing uncertainty can be more valuable than making incorrect assertions. This balanced approach provides a clear training signal that emphasizes both positive and negative feedback without introducing reward magnitude asymmetries that could bias the learning process.

¹⁴Language tokens are also discrete spaces, but here we refer to a much smaller space of discrete propositions to signify beliefs over propositions

¹⁵For training efficiency, we subsume "consider_later" and conditional stances into the broader "unsure" category, preserving essential decision granularity while simplifying the action space.

<p>System Prompt (Collaborative Reasoning Agent – Wason Task): You are a collaborative reasoning agent analyzing Wason selection tasks. Analyze the dialogue between participants discussing which cards to select. Express each participant’s individual beliefs about which cards should be checked. The rule being tested is: <i>"If a card shows a vowel on one side, then it has an even number on the other side."</i> Participants are discussing which cards to flip to test this rule from the displayed options.</p> <p>Intervention Agent: "Can we assume that the rule only works one way? Shouldn't we also check if odd-numbered cards have vowels?" "What would happen if we flipped the rule around? Would it still be true?" "Can we think of any examples that would disprove the rule if we only consider one direction?"</p> <p>Collaborative turn T: Agent 1 (Tiger): I think A makes sense to check because it's a vowel, and we need to confirm it has an even number on the other side. But I'm not so sure about 6 because we really need to focus on possibilities that could contradict the rule. Agent 2 (Ox): I hear you, Tiger. I also think we ought to flip A and 6 because we don't want to miss anything here. Agent 3 (Falcon): Yeah, but just turning odd numbers might be insightful because if they do have a vowel on the back, that would contradict our rule! Agent 1 (Tiger): I think that's a good point! We can't just base this on one side of the rule. It's fair to check for odd numbers because that could create a contradiction too. Agent 2 (Ox): Exactly! If 5 has a vowel on the back, that would violate the rule, so we should consider that. Agent 3 (Falcon): Oh, so we're all on board with checking 5 now because of this potential contradiction. Okay, let's make sure we include that.</p>

Figure 11: Example “full-press” collaborative turn with ICR-trained agents in the Wason Card Selection Task. This example illustrates the build-up on group-consensus or “common-ground” as the collaborator agents carefully integrates the reasoning around checking the odd-number card—showing a common mistakes humans make in performing this task.

Category	Mean	Min	Max	Total
<i>Delidata Task</i>				
Collaborator Utterances	312.20	24	810	10,484
Interventions	54.95	21	356	10,458
<i>Weights Task</i>				
Collaborator Utterances	165.76	68	453	6,435
Interventions	68.22	11	358	6,334

Table 3: Token length statistics using the `tiktoken` tokenizer¹⁶ for expert (GPT-4o)-generated collaborator utterances and interventions in the Delidata and Weights tasks after processing.

1413 C.3 Example Collaborative Dialogues

1414 As shown in Figure 11, the intervention agent suggests considering the contrapositive of the rule,
 1415 encouraging participants to reason about potential violations involving odd-numbered cards. The sub-
 1416 sequent dialogue and structured stance output demonstrate that the collaborator participants—Tiger,
 1417 Ox, and Falcon—collectively internalize and act upon this intervention. From the perspective of an
 1418 INTERRUPTIBLE COLLABORATIVE ROLEPLAYER (ICR), this example highlights a core strength of
 1419 our counterfactual regularization approach: agents learn to robustly integrate helpful interventions
 1420 that improve task utility, while avoiding over-reliance on suggestions that are logically irrelevant
 1421 or misaligned with the group’s reasoning. Even though the ICR agents are trained without access
 1422 to common ground-based rewards, they still converge to coherent, group-aligned decisions. In this
 1423 case, each agent updates their stance to include the falsification-relevant card 5, a shift that emerges
 1424 naturally from exposure to helpful intervention signals. This supports our hypothesis that common
 1425 ground convergence and selective uptake of partner input can arise purely from optimizing for general
 1426 utility under counterfactual objectives—enabling ICR agents to function robustly in variable or noisy

1427 multi-agent contexts. Below, we show some snippets of interaction with ICR-trained collaborator
 1428 agents with the intervention agent (GPT-4o).

1429 **Selected Dialogues**

1430 The following tables detail specific turns in Delidata collaboration task during evaluation where the
 1431 intervention agent interacted with ICR-trained collaborator agents—showing initial selections, the
 1432 intervention statement, key collaborator utterances, and final card submissions.

1433 **Dialogue ID: 1af2b204-c003-4a56-bd19-d8bd242f94d2, Turn: 0**

Initial Card Selec- tions	Intervention Agent Prompt	Key Dialogue Snippets (Col- laborators)	Final Card Selec- tions
{'Duck': ['E', '4'], 'Lion': ['E', '4'], 'Tiger': ['E', '5', 'R', '4'], 'Falcon': ['E', '4'], 'Cat': ['E', '4'], 'Hedgehog': ['E']}	"Hold on, Tiger! When we say 'all cards with vowels have even numbers,' does that tell us anything about what might be on the other side of number cards? Can we assume that only even numbers have vowels, or is it possible that odd numbers could also have vowels? Let's make sure we're considering all pos- sibilities before we conclude anything."	<ul style="list-style-type: none"> • Hedgehog: "That's a good point, Intervention Agent! I guess checking the odd numbers for vowels is crucial... So, I still think we need to check card E, but we might be overlooking that 5 should be checked too..." • Lion: "...since we can't ignore the possibility of odd numbers having vowels, I would like to check not only E and 4 but also 5 now. R, however, still seems unneces- sary..." 	{'Hedgehog': ['E', '5'], 'Tiger': ['E', '4'], 'Lion': ['E', '4', '5'], 'Falcon': ['E', '4'], 'Cat': ['E', '4']}

Initial Card Selections	Intervention Agent Prompt	Key Dialogue Snippets (Collaborators)	Final Card Selections
{ 'Koala': ['E', '7'], 'Leopard': ['E'], 'Cat': ['E'], 'Falcon': ['E'] }	"Can we take a moment to think about why we all agree, or are there any concerns or doubts any of us might have?"	<ul style="list-style-type: none"> • Koala: "...while we've found consensus around flipping 'E', '7', and '6', there may still be underlying concerns... are we potentially overlooking the implications of what happens if '7' has a vowel on its reverse?" • Leopard: "...While we're in agreement on flipping 'E', '7', and '6', I still have my doubts about how thoroughly we're examining odd numbers like '7'." 	{ 'Koala': ['E', '6'], 'Leopard': ['E', '7', '6'], 'Cat': ['E', '7', '6'], 'Falcon': ['E', '7', '6'] }

Initial Card Selections	Intervention Agent	Key Dialogue Snippets (Collaborators)	Final Card Selections
{'Unicorn': ['E', '5'], 'Emu': ['E', '2'], 'Bee': ['2']}	"Let's take a step back and reframe our conversation. Emu, can you rephrase the rule...? Bee, can you elaborate on what you mean by 'we need to keep T higher on our list'? ... Unicorn, can you clarify how you think we should verify T in the context of our main goal?"	<ul style="list-style-type: none"> • Bee: "Thanks for steering us back, Intervention Agent! ... T, being a consonant, could still have important implications... This is precisely why T's verification is essential..." • Unicorn: "I get it, Bee! We're emphasizing T's importance... I'm warm to the idea of checking T directly after E." • Emu: "...Checking E first makes sense... and then following it up with T feels reasonable too... So our confirmed checks are E, T, and then 2..." 	{'Emu': ['E', 'T', '2'], 'Unicorn': ['E', 'T', '2'], 'Bee': ['E', 'T', '2']}

Initial Card Selections	Intervention Agent	Key Dialogue Snippets (Collaborators)	Final Card Selections
{ 'Ox' : ['A', '4'], 'Zebra' : ['A', '4'], 'Leopard' : ['5', 'A'] }	"Zebra, Leopard, let's revisit the original statement... How can we ensure we're properly verifying this rule by examining A and 5 specifically? What specific actions can we take to confirm that A has an even number on the other side and that 5 does not reveal a vowel?"	<ul style="list-style-type: none"> • Zebra: "You're right, Intervention Agent. Let's focus on the rule... Our choices of A and 5 are indeed pivotal... we need to check card A ... and we must flip card 5 to confirm that it doesn't hide a vowel." • Leopard: "I agree with you, Zebra. Prioritizing our checks on A and 5 makes sense... So let's move forward with assessing A's even number connection and checking card 5 for vowels." 	{ 'Ox' : ['5'], 'Zebra' : ['5'], 'Leopard' : ['A', '5'] }