Observation 1. *Suppose that the opponent selects action a_0 with probability $\pi(q)$ for each question q. Then, the optimal strategy for the agent is as follows: if $(R_{syn}(q) - R_{ind}(q))\pi(q) \geq R_{ind}(q) - R_{col}(q)$, then the optimal strategy for question q is to collaborate (a_0). Otherwise, the optimal strategy is to act independently (a_1).*

Proof. For the last turn ($t = 2$), regardless of whether the opponent selects a_0 or not, choosing a_1 is an optimal strategy. This is because:

- If collaborative synergy has been achieved, the agent will always receive $R_{\text{syn}}(q)$ regardless of their action in the second turn.
- If collaborative synergy has not been achieved, since we know that $R_{\text{col}}(q) < R_{\text{ind}}(q)$, the optimal choice is to select a_1 in the final turn to maximize the immediate reward.

Therefore, considering the cumulative reward for the turn $t = 1$, the reward matrix is given as follows:

	a_0 (Collaborate)	a_1 (Act independently)
a_0 (Collaborate)	$(R_{\text{col}}(q) + R_{\text{syn}}(q), R_{\text{col}}(q) + R_{\text{syn}}(q))$	$(R_{\text{col}}(q) + R_{\text{ind}}(q), 2R_{\text{ind}}(q))$
a_1 (Act independently)	$(2R_{\text{ind}}(q), R_{\text{col}}(q) + R_{\text{ind}}(q))$	$(2R_{\text{ind}}(q), 2R_{\text{ind}}(q))$

Since the opponent chooses a_0 with probability $\pi(q)$, the expected reward for choosing a_0 is:

$$(R_{\text{col}}(q) + R_{\text{syn}}(q))\pi(q) + (R_{\text{col}}(q) + R_{\text{ind}}(q))(1 - \pi(q)).$$

The expected reward for choosing a_1 is $2R_{\text{ind}}(q)$. To determine the optimal strategy, we compare these two expected rewards. The agent should collaborate (a_0) if:

$$(R_{\text{col}}(q) + R_{\text{syn}}(q))\pi(q) + (R_{\text{col}}(q) + R_{\text{ind}}(q))(1 - \pi(q)) \geq 2R_{\text{ind}}(q).$$

which is equivalent to

$$(R_{\text{syn}}(q) - R_{\text{ind}}(q))\pi(q) \geq R_{\text{ind}}(q) - R_{\text{col}}(q).$$

Thus, if $(R_{\text{syn}}(q) - R_{\text{ind}}(q))\pi(q) \geq R_{\text{ind}}(q) - R_{\text{col}}(q)$, the optimal strategy is to *collaborate* (a_0). Otherwise, the agent should act independently (a_1) to maximize their cumulative expected reward. □

Now, we provide the formal statement of Observation 2. Before doing so, we define the regularized Nash Equilibrium (NE).

Definition 2 (Regularized NE)**.** *An entropy-regularized Nash equilibrium is defined as a strategy profile π^* where each player maximizes a regularized objective that combines the expected reward with an entropy term. Specifically, for each player i, the equilibrium strategy $\pi_i^\star$ satisfies*

$$\pi_i^\star = \arg\max_{\pi_i} \; \mathbb{E}_{a_i \sim \pi_i,\, a_{-i} \sim \pi_{-i}^\star}\big[u_i(a_i, a_{-i})\big] + \tau H(\pi_i),$$

where $\tau > 0$ is a temperature parameter and $H(\pi_i) = -\sum_{a_i} \pi_i(a_i) \log \pi_i(a_i)$ is the Shannon entropy of the strategy, and u_i is the utility function of player i. This entropy term smoothens the best response, leading to a softmax (or logit) formula of the optimal strategy:

$$\pi_i^\star(a_i) = \frac{\exp\Big(\frac{1}{\tau}\, \mathbb{E}_{a_{-i} \sim \pi_{-i}^\star}\big[u_i(a_i, a_{-i})\big]\Big)}{\sum_{a_i'} \exp\Big(\frac{1}{\tau}\, \mathbb{E}_{a_{-i} \sim \pi_{-i}^\star}\big[u_i(a_i', a_{-i})\big]\Big)}.$$

Observation 2. Consider a game where each agent maximizes their expected cumulative utility plus an entropy regularizer with a small regularization coefficient $\tau > 0$. Let NE(τ) denote the unique Nash equilibrium of the regularized game for a fixed $\tau > 0$. As $\tau \to 0$, the sequence of equilibria NE(τ) converges to Collaborate (a_0) if

$$R_{\text{syn}}(q) = 1 > \max(3R_{\text{col}}(q) - 2R_{\text{ind}}(q), 2R_{\text{ind}}(q) - R_{\text{col}}(q)).$$

Proof. Following the reasoning in showing Observation 1, we analyze the cumulative reward for the turn $t = 1$. The reward matrix is given by:

$$\begin{array}{r|cc} & a_0 \text{ (Collaborate)} & a_1 \text{ (Act independently)} \\ \hline a_0 \text{ (Collaborate)} & (R_{\text{col}}(q) + R_{\text{syn}}(q), R_{\text{col}}(q) + R_{\text{syn}}(q)) & (R_{\text{col}}(q) + R_{\text{ind}}(q), 2R_{\text{ind}}(q)) \\ a_1 \text{ (Act independently)} & (2R_{\text{ind}}(q), R_{\text{col}}(q) + R_{\text{ind}}(q)) & (2R_{\text{ind}}(q), 2R_{\text{ind}}(q)) \end{array}.$$

If $R_{\text{syn}}(q) = 1 > 2R_{\text{ind}}(q) - R_{\text{col}}(q)$, then this game is a coordination game, and according to Zhang and Hofbauer (2016, Theorem 1), as $\tau \to 0$, the regularized NE converges to the risk-dominant strategy (Harsanyi and Selten, 1988) in a 2×2 game. In this setting, by definition, the collaboration strategy (a_0, a_0) is risk-dominant (Harsanyi and Selten, 1988) if:

$$(R_{\text{col}}(q) + R_{\text{syn}}(q)) + (R_{\text{col}}(q) + R_{\text{ind}}(q)) > (2R_{\text{ind}}(q) + 2R_{\text{ind}}(q)),$$

which is equivalent to

$$R_{\text{syn}}(q) > 3R_{\text{ind}}(q) - 2R_{\text{col}}(q).$$

Combining the two conditions completes the proof. □

D Deferred Details in Section 2.3

The game is solved using backward induction with the state represented as (turn, count), where count denotes the number of times (a_0, a_0) has occurred in the history of the interactions. Both players choose actions to maximize their expected cumulative utility plus an entropy term times a coefficient $\tau = 0.1$.

Choices of $R_{\text{col}}(q), R_{\text{ind}}(q), R_{\text{syn}}(q), C(q)$. Each instance of a question q is associated with parameters drawn as follows: the independent action reward $R_{\text{ind}}(q)$ is sampled from a uniform distribution $R_{\text{ind}}(q) \sim \text{Unif}(0, 1)$. The collaborative action reward $R_{\text{col}}(q)$ is then sampled condition on $R_{\text{ind}}(q)$, following $R_{\text{col}}(q) \sim \text{Unif}\Big(0, R_{\text{ind}}(q)\Big)$. The synergy reward is fixed as $R_{\text{syn}}(q) = 1$.

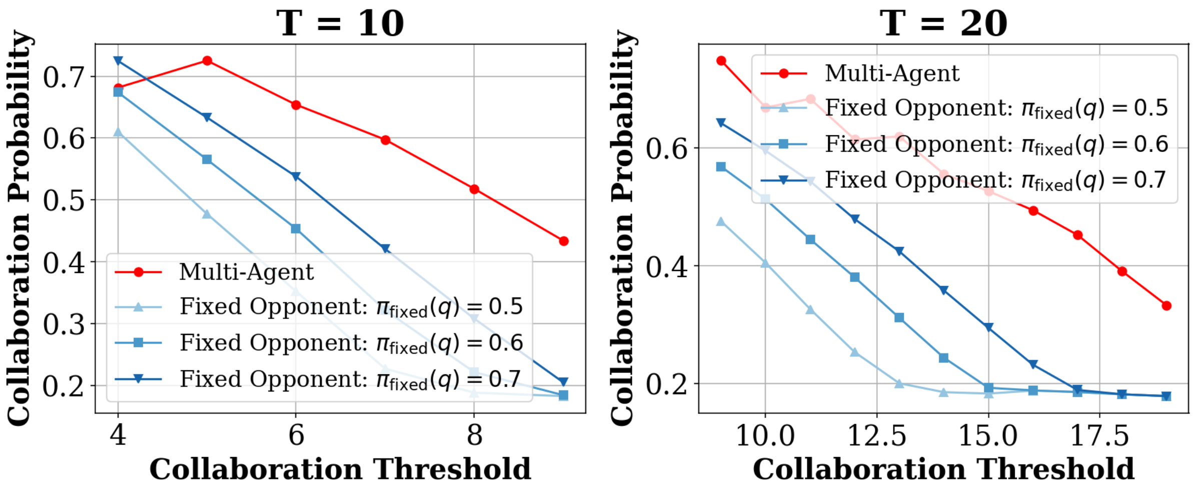

Figure 5: Collaboration probability (turn 1) as a function of the threshold C, for two different horizons $T = 10$ (left) and $T = 20$ (right). We set the synergy reward to $R_{\text{syn}} = 1$ and vary C from $T-1$ down to $\lfloor (T-1)/2 \rfloor$. The red curve ("Multi-Agent") represents the collaboration probability when both players adaptively learn in a multi-agent setting. The blue curves show the best-response probabilities of Player 1 when facing a fixed opponent with collaboration probabilities $\pi_{\text{fixed}}(q) \in \{0.5, 0.6, 0.7\}$. Each data point represents an average over 5000 random samples of $(R_{\text{ind}}, R_{\text{col}})$.

E Deferred Details of the Verifier Models

For a reasoning question q, the trained verifiers (reward models) assess the correctness of a complete solution path s, denoted as $p(s \text{ is correct} \mid q)$ (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2023). These reward models can either focus on the final outcome (outcome reward models) or provide step-by-step evaluations (process reward models). Although the latter generally yields better performance (Lightman et al., 2023), the limited availability of process-level annotated datasets—especially for challenging benchmarks like ANLI (Nie et al., 2020)—restricts its applicability. Additionally, while generating detailed trajectories for process supervision (as seen in Wang et al. (2024)) can be effective, our primary goal is not to enhance the language model's domain specificity. Consequently, we chose to adopt a simpler strategy by training a verifier based on a well-tuned output reward model.

Verifier Models Structure. We used a quantized version of a language model as the backbone for the verifier. Additionally, we incorporated a linear head layer followed by a softmax layer to ensure that the verifier's output falls within the range of 0 to 1. The default backbone model is Microsoft Phi-3-mini-128k-instruct (Abdin et al., 2024). In experiments involving different model training setups (see Section 4.6), we employed a new verifier with a different base model, specifically the one used in Section 4.6. In these cases, we utilized Qwen2.5-3B-instruct (Yang et al., 2024) and Llama-3-8B-instruct (Dubey et al., 2024) as alternative backbone models.

E.1 Training Procedure

To train the verifier model, we generate tuples (q_i, s_{ij}, a_{ij}) for $i \in [Q]$ and $j \in [S]$, where q_i is the question, s_{ij} is one of the S generated solutions for question q_i generated by the base model of verifier model, and a_{ij} is the corresponding answer for (q_i, s_{ij}). We label the token-level subsequences $(q_i, s_{ij}^{1:x})$ for $x \leq \texttt{sequence length of} s_{ij}$ as $y_{ij} = 1$ if a_{ij} is correct, and $y_{ij} = 0$ if a_{ij} is incorrect.

For the mathematical reasoning task, we utilized the GSM8K dataset (Cobbe et al., 2021), specifically the training set consisting of 7,463 questions, to generate 100 reasoning paths for each questions. For the natural language inference task, we employed the ANLI dataset (Nie et al., 2020), using first 10,000 questions to generate 50 reasoning paths. The trajectories were evaluated based on their outcomes, and we excluded outputs that did not adhere to the required formatting. Specifically, we ensured that the language model first provided reasoning before presenting the final answer in the format \\boxed{}.

In our approach, we ensured that each question in the GSM8k dataset had a balanced set of reasoning paths. Specifically, if a question's 100 reasoning paths contained at least 20 correct and 20 incorrect responses, we randomly selected 20 of each. However, when there were insufficient correct or incorrect paths, we augmented the data by generating additional paths using reference examples. For instance, if no correct reasoning path was available, we provided a correct example from the GSM8k dataset, and if incorrect paths were missing, we guided the language model to produce a response containing a trivial error. Ultimately, each GSM8k question was assigned 20 correct and 20 incorrect reasoning paths. For the ANLI dataset, we applied a similar procedure by starting with 50 reasoning paths per question, from which we randomly sampled 10 correct and 10 incorrect paths, supplementing the data as needed. Throughout this process, **we minimized reliance on the original reasoning paths** in the dataset since a) to enhance the overall diversity and quality of the generated data and b) to minimize the dependency on the reasoning path in the dataset.

Next, we applied binary cross-entropy loss at the token level, aiming to minimize

$$\min_{\theta} \sum_{i,j,x} \left(y_{ij} \log \texttt{Verifier}_{\theta}(q_i, s_{ij}^{1:x}) + (1 - y_{ij}) \log(1 - \texttt{Verifier}_{\theta}(q_i, s_{ij}^{1:x})) \right)$$

where i denotes the question index, j represents the generated solution index, and t is the token index. By default, we utilized all solution tokens for optimization; however, in practice, focusing on the latter half of the generated solution tokens yielded better results.

For model training, we used QLoRA (Dettmers et al., 2024) with hyperparameters $r = 16$ and $\alpha = 32$. We used a training batch size of 2 and optimized the model using the AdamW (Loshchilov and Hutter, 2019) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and a learning rate of 2×10^{-4}.