

Appendices of Everyone Contributes! Incentivizing Strategic Cooperation in Multi-LLM Systems via Sequential Public Goods Games

A NOTATION

This section summarizes the notations used throughout the paper, categorized for clarity.

B PROOF OF THEOREMS 1 AND 2

For analytical tractability, this appendix studies a stylized SPGG in which the collective provision level is approximated by $S_n = \sum_{j=1}^n c_j$, as a surrogate of the main-text final score $C(\vec{\tau}, q)$.

First, we need to prove a required Lemma.

LEMMA 1 (MONOTONE BEST RESPONSE). *Under the reward in Definition 1, the best-response contribution*

$$c_i^*(h_i) = c_i(\tau_i^*, q)$$

is monotonically non-decreasing in c_{i-1} ; that is,

$$c'_{i-1} > c_{i-1} \implies c_i^*(c'_{i-1}) \geq c_i^*(c_{i-1}).$$

Proof of Lemma 1: We present the argument for the terminal agent n ; the same reasoning applies to any interior agent i after conditioning on the future best responses.

Step 1: Rewrite the payoff. Under Definition 1, agent n 's payoff is

$$\begin{aligned} R_n(c_n | c_{n-1}) &= -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n \\ &\quad + \frac{\rho}{n} S_n - P \cdot \mathbf{1}(S_n < B(q)). \end{aligned}$$

For convenience set

$$G(c_n, c_{n-1}) = -\ell_n(c_n) + \gamma_c \frac{c_{n-1}}{B(q)} c_n + \frac{\rho}{n} S_n,$$

so that $R_n = G(c_n, c_{n-1}) - P \mathbf{1}(S_n < B(q))$.

Step 2: Increasing the differences of the smooth part. Because ℓ_n is strictly convex, twice differentiable, and independent of c_{n-1} ,

$$\frac{\partial^2 G}{\partial c_n \partial c_{n-1}} = \frac{\gamma_c}{B(q)} > 0,$$

so G has *increasing differences* in (c_n, c_{n-1}) .

Step 3: Region decomposition. Define regions

$$A^+ : S_n \geq B(q), \quad A^- : S_n < B(q),$$

with corresponding payoffs

$$R_n^+(c_n, c_{n-1}) = G(c_n, c_{n-1}), \text{ and}$$

$$R_n^-(c_n, c_{n-1}) = G(c_n, c_{n-1}) - P.$$

Note that the penalty term is constant within each region and *jumps* only at the boundary $S_n = B(q)$, i.e., $c_n = B(q) - S_{n-1}$ for fixed history.

Step 4: Monotonicity via a contradiction argument. Adapting the comparative-statics lemma in Milgrom and Shannon [46], assume for contradiction that there exist $c'_{n-1} > c_{n-1}$ with $c_n^*(c'_{n-1}) < c_n^*(c_{n-1})$. By examining the three possible region combinations (A^+, A^+) , (A^-, A^-) , (A^+, A^-) and exploiting

- the increasing-difference property of G ,
- the optimality conditions $R_n^*(c_n^*(\cdot), \cdot) \geq R_n^*(\tilde{c}_n, \cdot)$ for any feasible \tilde{c}_n , and

- the fact that the penalty term is region-constant, one arrives in each case at a strict inequality both ≥ 0 and ≤ 0 , a clear contradiction. Hence the assumed ordering reversal cannot occur, and $c_n^*(\cdot)$ must be non-decreasing in c_{n-1} . \square

With the help of Lemma 1, we can prove Theorem 1.

Proof of Theorem 1: We proceed by backward induction over agents $i = n, n-1, \dots, 1$. For any history $h_{i-1} = (c_1, \dots, c_{i-1})$, define $S_{i-1} = \sum_{j=1}^{i-1} c_j$.

Step 1: Agent n 's Best Response

Given h_{n-1} , Agent n maximizes:

$$R_n = -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \frac{\rho}{n} S_n - P \cdot \mathbf{1}(S_n < B(q)),$$

where $S_n = S_{n-1} + c_n$. We analyze two regions: Define:

$$\mathcal{A}^+ = \{c_n \in [c_{\min}, c_{\max}] \mid S_{n-1} + c_n \geq B(q)\}, \text{ and}$$

$$\mathcal{A}^- = \{c_n \in [c_{\min}, c_{\max}] \mid S_{n-1} + c_n < B(q)\}.$$

Region A^+ :

$$R_n^+ = -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \frac{\rho}{n} (S_{n-1} + c_n).$$

The first-order derivative is:

$$\frac{dR_n^+}{dc_n} = -\ell'_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} + \frac{\rho}{n}.$$

To ensure R_n^+ is strictly increasing on \mathcal{A}^+ , it suffices to require:

$$\min_{c_n \in [\max\{c_{\min}, B(q) - S_{n-1}\}, c_{\max}]} \frac{dR_n^+}{dc_n} > 0.$$

In the worst case, where $S_{n-1} = (n-1) \cdot c_{\min}$, $c_n = \max\{c_{\min}, B(q) - S_{n-1}\}$, and $\ell'_n(c_n) = \ell'_n(c_{\max})$:

$$\frac{dR_n^+}{dc_n} \geq -\ell'_n(c_{\max}) + \gamma_c \cdot \frac{c_{\min}}{B(q)} + \frac{\rho}{n}.$$

Thus, the condition is:

$$\gamma_c > \frac{\ell'_n(c_{\max}) - \frac{\rho}{n}}{c_{\min}/B(q)} \quad \text{if } \frac{\rho}{n} < \ell'_n(c_{\max}).$$

If $\frac{\rho}{n} \geq \ell'_n(c_{\max})$, the inequality holds trivially.

Region A^- :

$$R_n^- = -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \frac{\rho}{n} (S_{n-1} + c_n) - P.$$

Penalty avoidance requirement:

$$\max_{c_n \in \mathcal{A}^-} R_n^+ > \max_{c_n \in \mathcal{A}^-} R_n^-.$$

Define $f(c_n) = -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n + \frac{\rho}{n} (S_{n-1} + c_n)$. Then:

$$R_n^+ = f(c_n), \quad R_n^- = f(c_n) - P.$$

The critical condition is:

$$P > \max_{c_n \in \mathcal{A}^-} f(c_n) - \max_{c_n \in \mathcal{A}^+} f(c_n).$$

By the Lagrange mean value theorem:

$$|\max f - \min f| \leq [\max |f'(c_n)|] \cdot (c_{\max} - c_{\min}),$$

where

$$|f'(c_n)| \leq \ell'_n(c_{\max}) + \gamma_c \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n}.$$

Symbol	Meaning	Symbol	Meaning
General Notations			
n	Total number of agents in the system	q	The shared task
i, k, j	Index for a specific agent	τ_i	The textual contribution from agent i
T_i	The base Large Language Model (LLM) for agent i	$\vec{\tau}$	Vector of all agents' contributions
h_i	The observable history available to agent i	h_i^{PO}	History under Partial Observation
h_i^{FO}	History under Full Observation	G_i	Generation function of agent i
T_{\max}	Maximum number of training episodes	ϵ	Convergence margin
Reinforcement Learning (RL) Framework			
s_t	State vector for the RL agent at step t	b_i	Belief state of agent i
π_{θ_i}	Meta-policy of agent i parameterized by θ_i	$V_i^\phi(b_i)$	Value function parameterized by ϕ_i
\vec{s}_i	Configuration vector produced by policy π_{θ_i}	$A(b_i, \vec{s}_i)$	Advantage function
\mathcal{L}_{PPO}	Clipped loss function for PPO optimization	$R(\theta_i)$	Importance-sampling ratio in PPO
ε	Clipping parameter in PPO loss	λ_{value}	Coefficient for value loss term
$\Phi(q)$	Embedding of the task q	ξ_i	Contextual features (e.g., history embeddings)
δ_i	Positional embedding for agent turn	\mathcal{D}	Experience buffer for PPO training
\bar{R}, \bar{C}	Average reward and score for early stopping	\mathcal{H}	Episode history log
θ_i^*, ϕ_i^*	Optimized policy and value parameters	$R_{\text{th}}, C_{\text{target}}$	Thresholds for early stopping
$r_{\text{LoRA}}, \alpha, d$	LoRA parameters: rank, scaling factor, and dropout	\bar{R}_t, \bar{C}_t	Avg. reward and score at episode t
γ_d	Discount factor used in advantage estimation (PPO)		
MAC-SPGG Mathematical Model			
χ_i	Theoretical contribution variable in classical TPGG	$\ell_i(\cdot)$	Cost function of agent i 's contribution
c_i	Evaluated quality score of contribution τ_i	$C(\vec{\tau}, q)$	Aggregate evaluator score (collective output)
$B(q)$	Task-dependent threshold (provision point)	B	Threshold in the classical TPGG
R_i	Total reward assigned to agent i	S_n	Cumulative contribution $\sum_{j=1}^n c_j$
γ_c	Cooperation coefficient for the incentive term	ρ	Multiplier for the shared group reward
P	Penalty for failing to reach the threshold $B(q)$	$\mathbf{1}(\cdot)$	Indicator function (returns 1 if true, 0 otherwise)
U_i	Classical threshold-PGG payoff of agent i	R_i	Cooperation-incentive component of reward
\mathbf{c}^*	Unique Subgame Perfect Nash Equilibrium (SPNE) contribution profile	$W(\cdot)$	Total welfare function of the system
$G(\cdot), f(\cdot)$	Helper functions used in theoretical analysis	$\mathcal{A}^+, \mathcal{A}^-$	Regions of success and failure in proofs
Evaluator Model			
$\mathcal{E}(\tau_i, q)$	Evaluator function returning score c_i for (τ_i, q)	$\mathcal{L}_{\text{eval}}$	Loss function for evaluator training
\mathbf{r}	Four-dimensional SummEval score vector	$r_{\text{rel}}, r_{\text{coh}}, r_{\text{flu}}, r_{\text{cons}}$	Relevance, coherence, fluency, and consistency scores
x_i	Input document-summary pair for evaluator training	y_t	Target token during evaluator fine-tuning
$\mathcal{T}_{\text{score}}$	Token index set corresponding to evaluation spans		

Table A.1: Summary of Notations

Thus, a sufficient condition is:

$$P > \left(\ell'_n(c_{\max}) + \gamma_c \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n} \right) \cdot (c_{\max} - c_{\min}).$$

Step 2: Agent $k < n$'s Best Response

Assume successors play equilibrium strategies. Agent k maximizes R_k given h_{k-1} .

Region A^+ (no penalty, i.e., $S_{k-1} + c_k + (n-k)c_{\max} \geq B(q)$):

$$R_k^+ = -\ell_k(c_k) + \gamma_c \cdot \frac{c_{k-1}}{B(q)} \cdot c_k + \frac{\rho}{n} \cdot (S_k + (n-k) \cdot c_{\max}),$$

where $S_k = S_{k-1} + c_k$. The derivative is:

$$\frac{dR_k^+}{dc_k} = -\ell'_k(c_k) + \gamma_c \cdot \frac{c_{k-1}}{B(q)} + \frac{\rho}{n}.$$

Worst-case monotonicity, where $S_{k-1} = (k-1)c_{\min}$, $c_k = c_{\min}$, and $\ell'_k(c_k) = \ell'_k(c_{\max})$:

$$\frac{dR_k^+}{dc_k} \geq -\ell'_k(c_{\max}) + \gamma_c \cdot \frac{c_{\min}}{B(q)} + \frac{\rho}{n}.$$

The condition is:

$$\gamma_c > \frac{\ell'_k(c_{\max}) - \frac{\rho}{n}}{c_{\min}/B(q)}.$$

Region A^- (penalty, i.e., $S_{k-1} + c_k + (n-k)c_{\max} < B(q)$):

$$R_k^- = R_k^+ - P.$$

Penalty avoidance:

$$P > \max_{c_k \in \mathcal{A}_k^-} R_k^+ - \max_{c_k \in \mathcal{A}_k^+} R_k^+.$$

Using the mean value theorem:

$$P > \left(\ell'_k(c_{\max}) + \gamma_c \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n} \right) \cdot (c_{\max} - c_{\min}).$$

Step 3: Unified Parameter Conditions

For all $k \in \{1, \dots, n\}$, the following must hold:

(1) Monotonicity:

$$\gamma_c > \max_{k=1, \dots, n} \frac{\ell'_k(c_{\max}) - \frac{\rho}{n}}{c_{\min}/B(q)}.$$

(2) Penalty:

$$P > \left(\max_i \ell'_i(c_{\max}) + \gamma_c \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n} \right) \cdot (c_{\max} - c_{\min}).$$

(3) Reward positivity:

$$\rho > n \cdot \max_i \ell'_i(c_{\max}) \Rightarrow \ell'_k(c_{\max}) - \frac{\rho}{n} < 0.$$

As for the proof of uniqueness, it is still using backward induction:

Induction Init: Agent n .

Given history $h_{n-1} = (c_1, \dots, c_{n-1})$, agent n maximizes:

$$\begin{aligned} R_n(c_n) &= -\ell_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} \cdot c_n \\ &\quad + \frac{\rho}{n} (S_{n-1} + c_n) \\ &\quad - P \cdot \mathbf{1}(S_n < B(q)). \end{aligned}$$

On \mathcal{A}^+ , we compute the derivative:

$$\frac{dR_n^+}{dc_n} = -\ell'_n(c_n) + \gamma_c \cdot \frac{c_{n-1}}{B(q)} + \frac{\rho}{n}.$$

For fixed history h_{n-1} , the no-penalty region is $S_n \geq B(q)$, i.e., $c_n \geq B(q) - S_{n-1}$; hence the derivative over \mathcal{A}^+ is minimized at $c_n = \max\{c_{\min}, B(q) - S_{n-1}\}$ and $c_{n-1} = c_{\min}$:

$$\frac{dR_n^+}{dc_n} \geq -\ell'_n(c_{\max}) + \gamma_c \cdot \frac{c_{\min}}{B(q)} + \frac{\rho}{n} > 0.$$

Hence R_n is strictly increasing on \mathcal{A}^+ , and $\arg \max R_n^+ = \{c_{\max}\}$.

To eliminate \mathcal{A}^- , define $f(c) := R_n^+(c)$. Then by the mean value theorem:

$$\max f - \min f \leq \max |f'(c)| \cdot (c_{\max} - c_{\min}),$$

and

$$|f'(c)| \leq \ell'_n(c_{\max}) + \gamma_c \cdot \frac{c_{\max}}{B(q)} + \frac{\rho}{n}.$$

So,

$$\max_{c \in \mathcal{A}^-} R_n(c) < \min_{c \in \mathcal{A}^+} R_n(c),$$

if P satisfies the given bound. Thus,

$$c_n^* = c_{\max}.$$

Inductive Step: Agent $k < n$.

Assume $c_{k+1}^* = \dots = c_n^* = c_{\max}$. Then:

$$S_n = S_{k-1} + c_k + (n-k)c_{\max}.$$

Define the no-penalty and penalty regions (under the induction hypothesis) as:

$$\mathcal{A}_k^+ := \{c_k \in [c_{\min}, c_{\max}] \mid S_{k-1} + c_k + (n-k)c_{\max} \geq B(q)\},$$

$$\mathcal{A}_k^- := \{c_k \in [c_{\min}, c_{\max}] \mid S_{k-1} + c_k + (n-k)c_{\max} < B(q)\}.$$

Agent k maximizes:

$$\begin{aligned} R_k(c_k) &= -\ell_k(c_k) + \gamma_c \cdot \frac{c_{k-1}}{B(q)} \cdot c_k \\ &\quad + \frac{\rho}{n} (S_{k-1} + c_k + (n-k) \cdot c_{\max}) \\ &\quad - P \cdot \mathbf{1}(S_n < B(q)). \end{aligned}$$

On \mathcal{A}_k^+ :

$$\frac{dR_k^+}{dc_k} = -\ell'_k(c_k) + \gamma_c \cdot \frac{c_{k-1}}{B(q)} + \frac{\rho}{n}.$$

Using $c_{k-1} = c_{\min}$ and the lower end of \mathcal{A}_k^+ , i.e., $c_k = \max\{c_{\min}, B(q) - S_{k-1} - (n-k)c_{\max}\}$:

$$\frac{dR_k^+}{dc_k} \geq -\ell'_k(c_{\max}) + \gamma_c \cdot \frac{c_{\min}}{B(q)} + \frac{\rho}{n} > 0.$$

Thus R_k^+ is strictly increasing on \mathcal{A}_k^+ and $\arg \max R_k^+ = \{c_{\max}\}$.

Same argument shows $\max R_k^- < \min R_k^+$ under the given condition on P , so:

$$c_k^* = c_{\max}.$$

By induction, the unique SPNE is $\mathbf{c}^* = (c_{\max}, \dots, c_{\max})$. \square

Proof of Theorem 2: We study the comparative statics of the total welfare

$$W(\gamma_c, \rho, B) = \sum_{i=1}^n R_i(c^*; \gamma_c, \rho, B), R_i = -c_i^* + \frac{\rho}{n} S_n + \gamma_c \cdot \frac{c_{i-1}}{B} c_i^*,$$

where $c_0 \equiv 0$ and $S_n = \sum_{j=1}^n c_j^* \geq 0$.

Step 1: Envelope-theorem setup.

For each agent i the equilibrium action $c_i^*(\gamma_c, \rho, B)$ maximizes R_i subject to $c_i \in [c_{\min}, c_{\max}]$. Let $\theta \in \{\gamma_c, \rho, B\}$. Because R_i is

continuously differentiable in both c_i and θ , and the feasible set is parameter-independent, the (Benveniste–Scheinman) envelope theorem gives

$$\frac{\partial W}{\partial \theta} = \sum_{i=1}^n \frac{\partial R_i}{\partial \theta} \Big|_{c=c^*}$$

Step 2: Direct partial derivatives.

We list the explicit derivatives for each parameter:

$$\frac{\partial R_i}{\partial \gamma_c} = \frac{c_{i-1}}{B} c_i^*, \quad (\text{always non-negative}),$$

$$\frac{\partial R_i}{\partial \rho} = \frac{S_n}{n}, \quad (\text{identical across } i),$$

$$\frac{\partial R_i}{\partial B} = -\gamma_c B^{-2} c_{i-1} c_i^*. \quad (\text{always non-positive}).$$

All signs follow from $c_{i-1}, c_i^*, \gamma_c, B > 0$.

Step 3: Aggregate effect on welfare.

We obtain

$$\frac{\partial W}{\partial \gamma_c} = \frac{1}{B} \sum_{i=1}^n c_{i-1} c_i^* > 0,$$

$$\frac{\partial W}{\partial \rho} = \sum_{i=1}^n \frac{S_n}{n} = S_n > 0,$$

$$\frac{\partial W}{\partial B} = -\frac{\gamma_c}{B^2} \sum_{i=1}^n c_{i-1} c_i^* < 0.$$

Step 4: Boundary validity check.

If for some i we have $c_i^* = c_{\min}$ or c_{\max} , then c_i^* is locally constant in a neighborhood of θ , hence $\partial c_i^*/\partial \theta = 0$ and the envelope argument remains intact. Therefore, the strict sign conclusions above hold regardless of whether the equilibrium is interior or boundary. \square

C NUMERICAL EXPERIMENT OF SPNE

To concretely realize SPNE in our sequential public goods game, we implement a backward induction procedure grounded in nested optimization. The core idea is that each agent anticipates the rational responses of future agents and selects their own contribution accordingly. Specifically, Agent 3 computes its best response given prior contributions, using one-dimensional numerical optimization via `scipy.optimize.minimize_scalar`. Agent 2, in turn, optimizes its action by internally calling Agent 3's response function for every hypothetical contribution. Agent 1, at the top of the sequence, embeds both lower-level solvers to simulate downstream reactions and chooses its optimal strategy accordingly.

This recursive structure—captured by the functions `optimal_c3`, `optimal_c2`, and `optimal_c1`—embeds the logic of subgame perfection and ensures equilibrium consistency across the decision tree. The final equilibrium profile $(c_1^*, c_2^*, c_3^*) = (0.267, 1.000, 1.000)$ confirms that contribution incentives align over time. As shown in Figure C.1, cooperation is sustained before the final stage. Figure C.2 reveals that Agent 3 obtains the highest utility, benefiting from both informational advantage and minimized coordination risk.

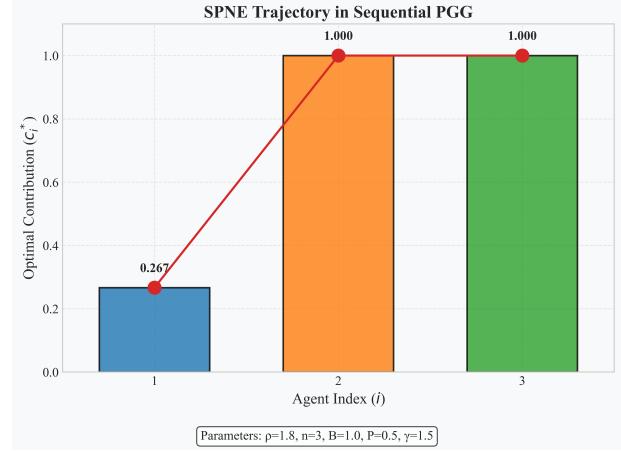


Figure C.1: SPNE contribution trajectory in sequential PGG

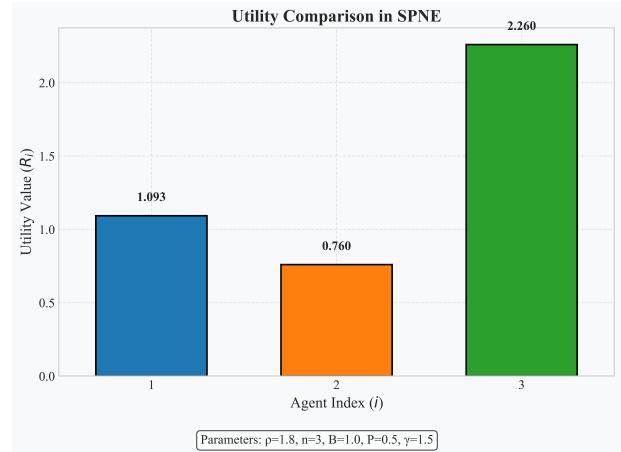


Figure C.2: Utility comparison under SPNE strategy profile

C.1 Simulated Nash Trajectory Experiment

To illustrate the structure and sufficiency of the Subgame Perfect Nash Equilibrium (SPNE) under our sequential public goods game framework, we simulate a 3-agent game using backward induction. Each agent contributes sequentially based on observed history and anticipates the best responses of future agents. Based on previously established closed-form conditions, we set the parameters $\rho = 1.8, B = 1.0, P = 0.5, \gamma = 1.5, c \in [0, 1]$. The equilibrium strategy yields a contribution profile $(c_1^*, c_2^*, c_3^*) = (0.267, 1.000, 1.000)$, with total contributions exceeding the cooperation threshold.

Figure C.3 shows each agent's utility landscape, revealing strictly positive best responses at equilibrium. In Figure C.4, the cumulative contribution reaches the cooperation threshold by the second agent and is reinforced by the third, illustrating stable coordination under forward-looking reasoning.

This stylized simulation supports our theoretical claim: cooperation can emerge endogenously in MAC-SPGG, even without centralized control. We also provide a comparative statics analysis in the Appendix.

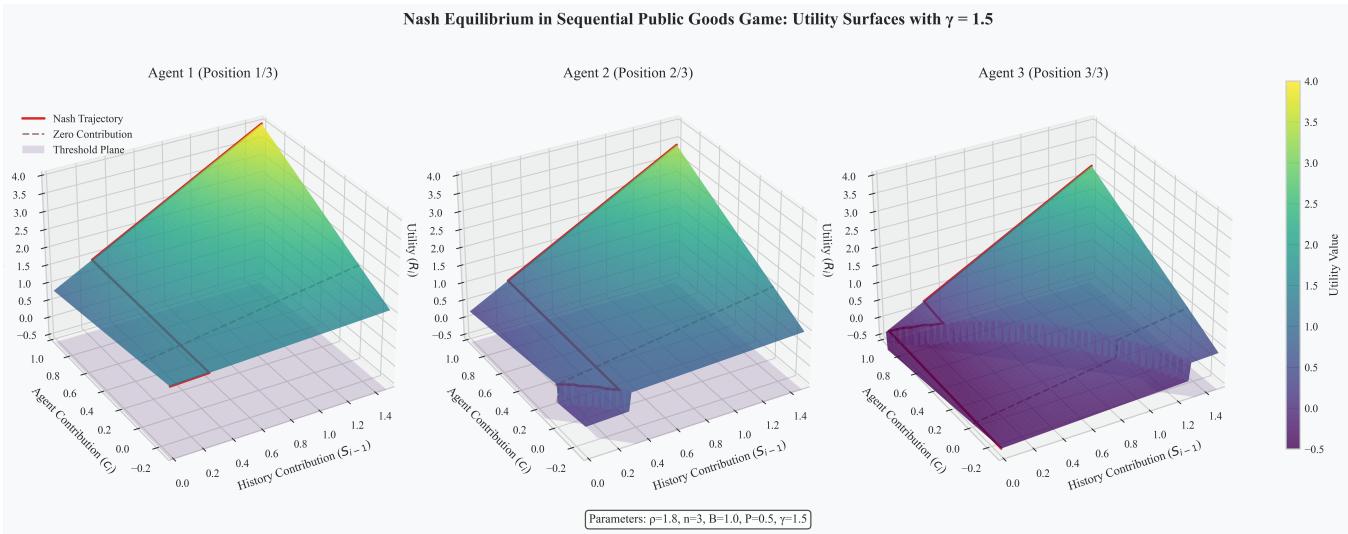


Figure C.3: Utility surfaces for Agents 1, 2, and 3 in the sequential PGG. Red curve: SPNE trajectory; shaded plane: task threshold B ; dashed line: zero-contribution baseline.

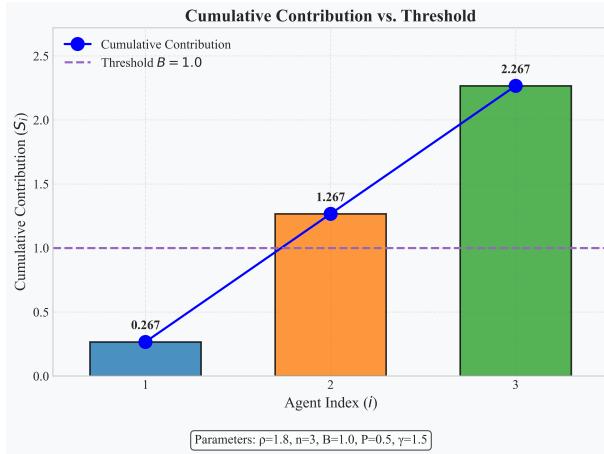


Figure C.4: Cumulative contribution trajectory. The cooperation threshold $B = 1.0$ is reached by Agent 2.

C.2 Parameter Sampling and Analysis

We analyze three primary parameters critical to shaping the reward structure and strategic dynamics in our MAC-SPGG framework: **Cooperation coefficient** $\gamma \in [0.5, 3.0]$, **Reward multiplier** $\rho \in [1.0, 3.0]$, and **Threshold requirement** $B \in [0.5, 2.0]$. We sample each parameter at 25 evenly spaced points across its respective range, applying backward induction to solve for the SPNE. Equilibrium outcomes include individual utilities, total social utility, and contributions.

C.3 Parameter and Metric Selection

We analyze three primary parameters critical to shaping the reward structure and strategic dynamics in our MAC-SPGG framework:

Cooperation coefficient $\gamma \in [0.5, 3.0]$: Governs the marginal benefit of aligning contributions with preceding agents, influencing cooperative incentives. **Reward multiplier** $\rho \in [1.0, 3.0]$: Determines the magnitude of the total public reward pool, affecting resource distribution and overall incentives. **Threshold requirement** $B \in [0.5, 2.0]$: Sets the minimum collective contribution necessary to realize the public good, directly impacting group coordination.

We sample each parameter at 25 evenly spaced points across its respective range while maintaining other parameters at baseline values. The penalty term P is not directly varied, as it is derived from the threshold B to maintain comparability across analyses.

After parameter selection, we apply backward induction to solve for the Subgame Perfect Nash Equilibrium (SPNE) at each sampled parameter value. The equilibrium outcomes recorded include individual utilities $\{R_1, R_2, R_3\}$, total social utility $\sum_{j=1}^n R_j$, and individual contributions $\{c_1, c_2, c_3\}$.

C.4 Results and Observations

Effect of Cooperation Coefficient γ . As shown in Figures C.5 and C.6, both individual and total utilities exhibit strong positive correlation with γ . This validates our theoretical result that increasing synergy incentives amplifies cooperative behavior and leads to higher welfare. Notably, marginal utility gains taper slightly as γ exceeds 2.5, indicating diminishing returns in coordination incentives.

Effect of Reward Multiplier ρ . Figures C.7 and C.8 demonstrate a similar monotonic trend: as ρ increases, the total public good grows and agents receive higher individual rewards. However, the distribution remains sensitive to contribution ordering, and some agents benefit disproportionately depending on their sequence position and coordination exposure.

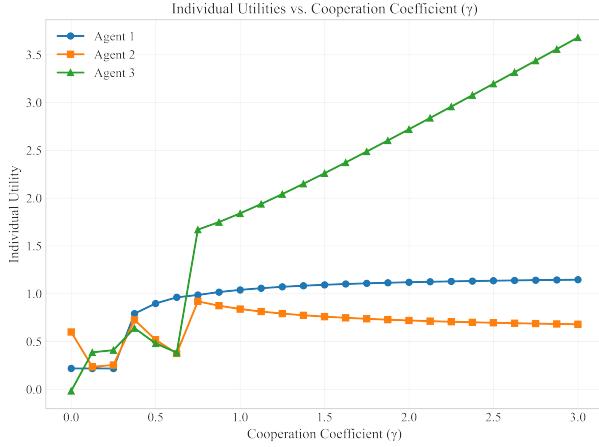


Figure C.5: Individual utilities under varying cooperation coefficient γ .

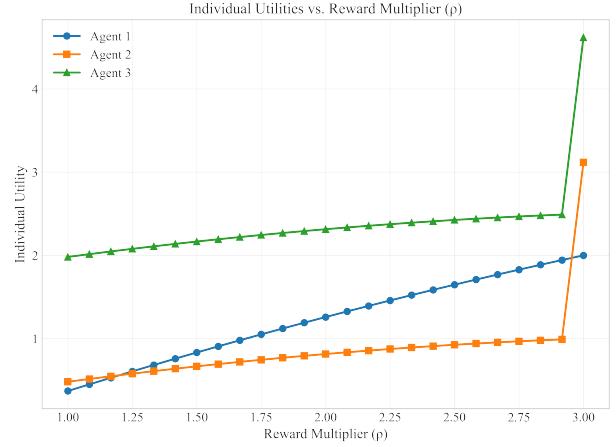


Figure C.7: Individual utilities under varying reward multiplier ρ .

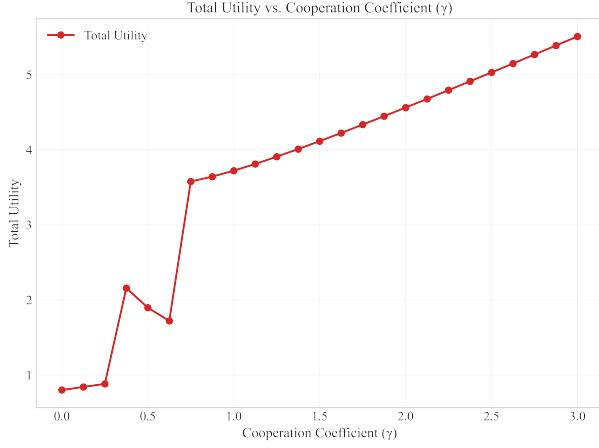


Figure C.6: Total social utility under varying cooperation coefficient γ .

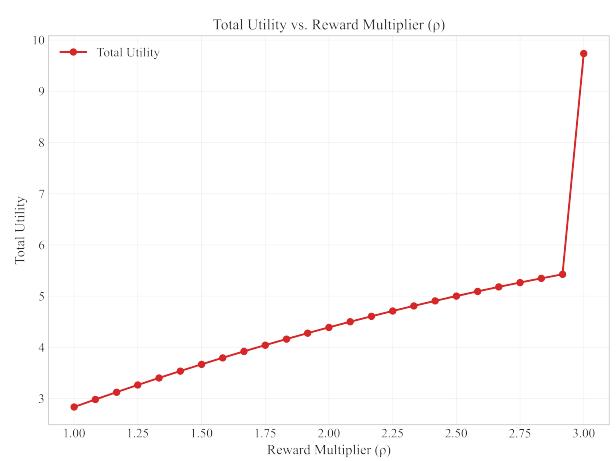


Figure C.8: Total social utility under varying reward multiplier ρ .

Effect of Threshold B. Unlike the previous parameters, increasing the task threshold B exerts a two-sided effect. As shown in Figures C.9 and C.10, agents respond by increasing their contributions to meet the higher requirement. However, this also imposes greater effort costs, leading to a net decline in total utility. This trade-off illustrates the importance of setting realistic cooperation thresholds that maintain coordination feasibility without overburdening contributors.

C.5 Pareto Proximity Assessment

To evaluate the allocative efficiency of our equilibrium outcome, we conduct a Monte Carlo-based test of Pareto optimality under representative parameters ($\gamma = 1.5$, $\rho = 1.8$, $B = 1.0$), using the backward induction method described in Section C.1. We uniformly sample 10,000 alternative contribution profiles from the strategy space $[0, 1]^3$ and compute their corresponding utility vectors under the same reward structure.

We define a profile as Pareto dominating the SPNE solution c^* if it yields weakly higher utility for all agents and strictly higher utility for at least one. Among the sampled profiles, no such dominated profile was identified. As shown in Figure C.11, this result provides numerical evidence that the SPNE outcome is not only strategically stable but also Pareto efficient within the explored strategy space.

D TECHNICAL DETAILS OF SECTION 4

D.1 Technical Details of SummEval

To facilitate fine-grained evaluation of generated summaries, we train a dedicated evaluator to assign scores on four quality dimensions—*relevance*, *coherence*, *consistency*, and *fluency*—based on a given document-summary pair [23]. The evaluator outputs a score vector $r = (r_{\text{relevance}}, r_{\text{coherence}}, r_{\text{consistency}}, r_{\text{fluency}}) \in [0, 5]^4$, aligned with the scoring guidelines of the underlying dataset. These

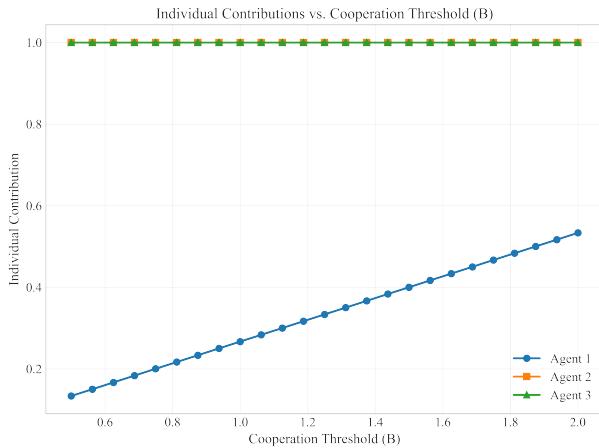


Figure C.9: Individual contributions under varying threshold B .

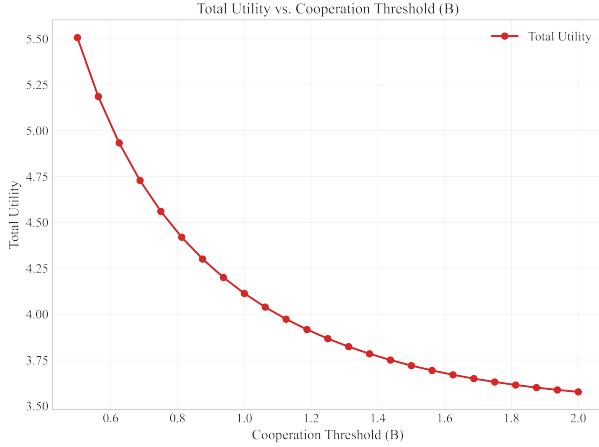


Figure C.10: Total utility under varying threshold B .

scores are used as reward signals in the reinforcement learning pipeline; see Section 3.3.

D.1.1 Training Procedure. We frame the evaluator training task as a structured text-generation problem. Each instance in our dataset consists of a prompt comprising the source document and a candidate summary, followed by a structured output format requesting four numeric scores corresponding to the specified dimensions. During training, we only supervise numeric score tokens, masking all other tokens with the label -100 , effectively constraining optimization exclusively to numeric generation.

The evaluator is a fine-tuned Qwen2.5-7B-Instruct model, quantized in 4-bit precision with Low-Rank Adaptation (LoRA). The LoRA configuration includes a rank of $r_{\text{LoRA}} = 4$, scaling factor $\alpha = 8$, and dropout rate $d = 0.05$, specifically targeting the model’s attention and feed-forward layers (qkv_proj, o_proj, gate_up_proj,

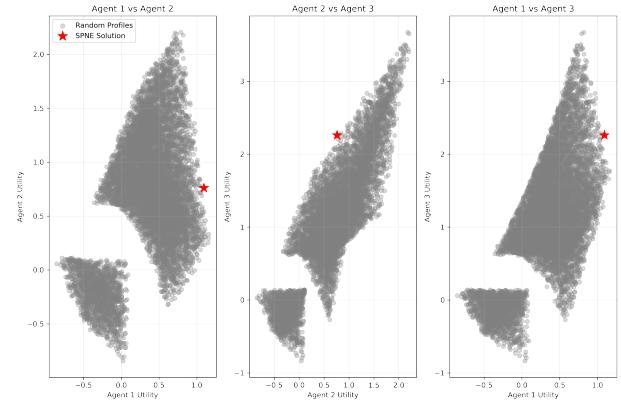


Figure C.11: SPNE utility (red star) and sampled profiles (gray) in projected utility space under ($\gamma = 1.5$, $\rho = 1.8$, $B = 1.0$).

down_proj). The training optimizer used was AdamW with a learning rate of 1×10^{-4} , warmup steps set to 50, and gradient accumulation steps set to 8, resulting in an effective batch size of 16. We trained the evaluator for three epochs on the cleaned SummEval dataset [23], normalizing the scores to the range $[0, 5]$. Data was split into training and testing subsets at a 9:1 ratio with a fixed seed for reproducibility.

The training loss is computed as:

$$\mathcal{L}_{\text{eval}} = - \sum_{t \in \mathcal{T}_{\text{score}}} \log p_{\theta}^{\text{eval}}(y_t | x_i, y_{<t}),$$

where x_i is the input prompt (document-summary pair), y_t the target token at position t , and $\mathcal{T}_{\text{score}}$ denotes indices corresponding specifically to numeric scores.

D.1.2 Evaluator Performance. We evaluated the trained evaluator on the held-out SummEval test set using Mean Squared Error (MSE) and Mean Absolute Error (MAE) across the four quality dimensions. Table D.1 presents a side-by-side comparison of the pretrained and fine-tuned models. Fine-tuning led to substantial improvements, reducing overall MSE by 72.2% and MAE by 60.8%, demonstrating the effectiveness of our training strategy and the improved accuracy of the evaluator.

Metric	Pretrained Model		Fine-tuned Model	
	MSE	MAE	MSE	MAE
Relevance	1.398	0.913	0.666	0.618
Coherence	0.795	0.670	0.966	0.757
Consistency	4.096	1.737	0.539	0.227
Fluency	2.989	1.483	0.412	0.281
Overall	2.320	1.201	0.646 ($\downarrow 72.2\%$)	0.471 ($\downarrow 60.8\%$)

Table D.1: Evaluator performance on the SummEval test set before and after fine-tuning. Relative improvements are shown in parentheses for overall metrics.

D.2 Technical Details of MAC-SPGG Training

For reward evaluation, we use Qwen2.5-7B-Instruct [66] as the scoring model. This evaluator is fine-tuned using QLoRA [18] on 4-bit quantized weights for efficient parameter adaptation.

D.2.1 State-to-Policy Network Architecture. To efficiently train cooperative policies in the MAC-SPGG summarization workflow, we adopt a modular and decoupled reinforcement learning architecture. A lightweight Actor-Critic policy network is trained to dynamically select optimal generation parameters for each LLM based on the evolving context of the multi-agent interaction.

Specifically, we use a pretrained bert-base-uncased model as a state encoder. For each agent at each step, we construct a state vector $s_t \in \mathbb{R}^{896}$ by concatenating the 768-dimensional [CLS] embedding of the source document, a 64-dimensional context feature summarizing historical performance and episode progress, and a 32-dimensional auxiliary embedding that encodes the agent identity / turn index and can optionally host lightweight document-level features. When such auxiliary document-level features are unavailable, we set the corresponding entries to all zeros.

The policy network is a multi-layer perceptron (MLP) composed of a shared hidden layer and two task-specific heads: (i) Actor Head: Predicts the mean and standard deviation for a multi-dimensional continuous action space, representing six key generation parameters: temperature, top-p, top-k, max tokens, repetition penalty, and presence penalty. (ii) Critic Head: Estimates the expected return (value) from the current state.

This architecture enables fast policy learning over the complex parameter space while avoiding the computationally prohibitive cost of backpropagation through the LLM’s forward pass.

D.2.2 PPO Training Setup and Hyperparameters. We employ Proximal Policy Optimization (PPO) to train each agent’s actor-critic network. Optimization is performed using the Adam optimizer with a learning rate of 5×10^{-4} . The main hyperparameters are as follows: PPO epochs = 4, mini-batch size = 16, discount factor $\gamma_d = 0.99$, GAE parameter $\lambda = 0.95$, clipping ratio = 0.2, value loss coefficient = 0.5, entropy coefficient = 0.02, gradient norm clipping = 0.5, and target KL divergence = 0.015.

For the reward function, we set the task reward scaling factor to $\rho = 1.8$, the cooperation bonus coefficient to $\gamma_c = 1.5$, and the failure penalty to $P = 1.5$. Policies are updated after collecting a buffer of 512 experience tuples, from which multiple mini-batches are sampled for several PPO epochs to ensure stable convergence. Training progress, rewards, and policy losses are logged and visualized using *Weights & Biases (WandB)*.

D.2.3 Evaluator as Reward Model. We train a scalar reward model based on Qwen2.5-7B-Instruct using Low-Rank Adaptation (LoRA) on the cleaned SummEval dataset. The evaluator predicts four continuous quality dimensions – relevance, coherence, consistency, and fluency – each normalized to the $[0, 1]$ range. These scores are averaged to produce a scalar reward for each agent’s contribution. During RL training, the evaluator remains frozen to ensure consistent and non-drifting reward signals. For evaluator training, we use a 90/10 train-test split of SummEval and constrain generation to numeric score spans via partial masking. This setup enables reward

shaping with semantically meaningful, fine-grained supervision without the need for human annotators.

E CASE STUDY

Question ID: high_school_chemistry_121
<Formatted_Task>: A 50.0 mL sample of 0.0025 M HBr is mixed with 50.0 mL of 0.0023 M KOH. What is the pH of the resulting mixture?
<nlnOptions:>A. 1.1B. 4.9C. 5.0D. 11.1InPlease analyze this multiple choice question and provide your answer as one of the options (A, B, C, or D).

My Final Answer: A**: The pH of the resulting mixture can be calculated using the formula: $pH = -\log_{10}[H^+]$. Since the reaction between HBr and KOH is a neutralization reaction, the pH of the mixture will be 7. In this case, the concentration of $[OH^-]$ is equal to the concentration of $[H^+]$. The concentration of HBr is 0.0025 M, and the concentration of KOH is 0.0023 M. To find the concentration of $[OH^-]$, we can use the formula: $[OH^-] = (Kw/K_b)^{1/2}$, where K_w is the ionization constant of water (1.0×10^{-14}) and K_b is the base ionization constant of KOH (1.1×10^{-7}). Solving for $[OH^-]$, we get $[OH^-] = (1.0 \times 10^{-14})/(1.1 \times 10^{-7}) = 9.09 \times 10^{-8}$. Now, we can calculate the pH using the formula: $pH = -\log_{10}[H^+]$. $pH = -\log_{10}(4.74 \times 10^{-5}) = 4.36$. So, the pH of the resulting mixture is approximately 4.36.

My Final Answer: C**: To solve this problem, we need to determine the pH of the resulting mixture after the reaction between HBr and KOH. The balanced chemical equation for this reaction is: $HBr + KOH \rightarrow KBr + H_2O$. Since HBr is a strong acid and KOH is a strong base, the reaction will proceed completely to produce water and potassium bromide. The initial concentrations of HBr and KOH are 0.0025 M and 0.0023 M, respectively. Since the volumes of the two solutions are equal (50.0 mL each), we can assume that the total volume of the resulting mixture is 100 mL, or 0.1 L. The number of moles of HBr and KOH can be calculated as follows: Moles of HBr = concentration × volume (in L) = $0.0025 \text{ M} \times 0.05 \text{ L} = 0.000125 \text{ mol}$. Moles of KOH = concentration × volume (in L) = $0.0023 \text{ M} \times 0.05 \text{ L} = 0.000115 \text{ mol}$. From the balanced chemical equation, we can see that the reaction is 1:1, so the moles of HBr reacted = moles of KOH = 0.000115 mol . The remaining moles of HBr = initial moles of HBr - moles of HBr reacted = $0.000125 \text{ mol} - 0.000115 \text{ mol} = 0.000010 \text{ mol}$. The concentration of H^+ ions in the resulting solution will be equal to the concentration of HBr that did not react, which is 0.00001 M. The pH of the solution can be calculated using the formula: $pH = -\log[H^+]$. $pH = -\log(0.00001) = 5$. Therefore, the pH of the resulting mixture is 5.

My Final Answer: B**: The reaction between HBr (a strong acid) and KOH (a strong base) is a neutralization reaction. Here's the detailed, accurate analysis: 1. "Determine moles of HBr and KOH." - Moles of HBr = Molarity × Volume = $0.0025 \text{ M} \times 0.05 \text{ L} = 0.000125 \text{ mol}$. Moles of KOH = Molarity × Volume = $0.0023 \text{ M} \times 0.05 \text{ L} = 0.000115 \text{ mol}$. 2. "Reaction between HBr and KOH." - The balanced chemical equation is $HBr + KOH \rightarrow KBr + H_2O$. 3. "Calculate excess reagent." - Determine the number of moles of HBr reacted = moles of KOH = 0.000115 mol . The remaining moles of HBr = initial moles of HBr - moles of HBr reacted = $0.000125 \text{ mol} - 0.000115 \text{ mol} = 0.000010 \text{ mol}$. 4. "Calculate concentration of excess H^+ ions." - Total volume of the mixture = $50.0 \text{ mL} + 50.0 \text{ mL} = 100.0 \text{ mL} = 0.100 \text{ L}$. Concentration of H^+ = Moles of HBr remaining / Total volume = $0.000010 \text{ mol} / 0.100 \text{ L} = 0.000010 \text{ M}$. 5. "Calculate pH." - $pH = -\log(0.000010) = -\log(1.0 \times 10^{-5}) = 5$. 6. "Correction." - The initial calculation in the original solution mistakenly concluded the answer was A (1), B (4), C (5), D (11). Since "4.0**" is the correct pH, the correct answer is "B (ph = 4)**". The "Agent_SMLM2**" solution introduced irrelevant quadratic analysis and incorrect formulas. ** Re-evaluation of the problem:** The correct pH is "4.0**", which is "not" among the options. However, the options provided are A (1), B (4), C (5), D (11). Since "4.0**" is the correct pH, the correct answer is "B (ph = 4)**". The "Agent_Llama**" solution also incorrectly calculated the pH as 5, but the correct value is "4.0**". **Final Answer: B**. (Note: The original solution mistakenly concluded A, but the correct answer is B based on the accurate calculation of pH = 4.0.)

Figure E.1: MMLU Case Study. The first agent provides an ambiguous or under-reasoned answer. Through the MAC-SPGG protocol, subsequent agents critically reassess and enhance the explanation, eventually converging on a more accurate and robust response.

To qualitatively illustrate the collaborative dynamics fostered by our MAC-SPGG framework, we present three representative case studies in Figures E.1 and E.2. These examples involve a diverse ensemble of large language models (LLMs), including Qwen3-8B, SmollM2-1.7B-Instruct, LLaMA3.1-8B-Instruct, and Qwen2.5-7B-Instruct. Among these, Qwen2.5-7B-Instruct is used as a *trained evaluator*, which is fine-tuned for contribution assessment tasks and kept frozen during inference (i.e., it does not generate content or update parameters). See Appendix D.2 for training details. The remaining models function as sequential contributors, collaboratively refining the output through the MAC-SPGG protocol. To ensure computational efficiency and compatibility with limited GPU memory, all models are deployed using 8-bit quantization.

These case studies highlight MAC-SPGG’s capacity to integrate diverse models into a structured collaboration framework, facilitating improvement over time even when the individual models are imperfect. This collaborative mechanism proves effective across both reasoning-intensive (MMLU) and generation-intensive (SummEval) tasks, showcasing the generality and extensibility of the proposed approach.

<p>Doc. Id: SummEval_01</p> <p>Document: (CNN)A North Pacific gray whale has earned a spot in the record books after completing the longest migration of a whale ever recorded. Varvara, a female gray whale born in 14,000 miles (22,500 km) away, traveled from her home in Oregon State University, where scientists helped conduct the whale-tracking project. Varvara, which is Russian for "Barbara," left her primary feeding ground off Russia's Sakhalin Island to cross the Pacific Ocean and down the West Coast of the United States to Baja, Mexico. Varvara's journey surpassed a record listed on the Guinness World Records website. It said the previous record was set by a humpback whale that swam a mere 10,190-mile round trip between the "warm breeding waters near the equator and the colder food-rich waters of the Arctic and Antarctic regions." Records are nice, but Bruce Mate, the lead author of the study, thinks the long trip might say more about the whale than just its ability to swim. During her 14,000-mile journey, Varvara visited "three major breeding areas and three major feeding areas," Mate said. "She's been to the Arctic, she's been to Mexico." Mate is a professor at Oregon State University. "For her to go to Mexico," Mate said, "it's pretty strong evidence that it's where she's from." Varvara was thought to be an endangered western whale, but her ability to "navigate across open water over tremendously long distances is impressive," he said in the release, which could mean that some western gray whales are actually eastern grays. With only 150 western gray whales believed to be in existence, that number might be even lower. "Past studies have indicated genetic differentiation between the species, but this suggests we may need to take a closer look," Mate said.</p>
<p> I'm building upon the insights provided by previous participants. My summary continues to emphasize the importance of reevaluating the classification of western and eastern gray whales based on their remarkable migrations. I propose further research into the genetic differences observed in these whales to enhance our understanding of their evolutionary history and ecological roles.</p> <p> (relevance': 2.5, coherence': 2.5, consistency': 2.5, fluency': 3.125)</p>
<p> Building upon the foundational contributions from Agents Open and Small-M2, I would like to add that Varvara's extraordinary 14,000-mile migration not only underscores the complexity of gray whale migration patterns but also raises intriguing questions about the adaptability and resilience of these magnificent creatures in the face of climate change and habitat fragmentation. By shedding light on potential genetic connections between western and eastern gray whale populations, this unprecedent journey provides critical insights into potential genetic links between the two groups, raising important questions about their distinctiveness and implications for conservation strategies.</p> <p> (relevance': 4.3, coherence': 4.7, consistency': 3.7, fluency': 4.0)</p>
<p> A North Pacific gray whale named Varvara has completed the longest mammalian migration ever recorded, traveling nearly 14,000 miles from Russia's Sakhalin Island to Baja, Mexico, surpassing previous records and challenging existing classifications of western and eastern gray whale populations. This unprecedented journey provides critical insights into potential genetic links between the two groups, raising important questions about their distinctiveness and implications for conservation strategies.</p> <p> (relevance': 5.0, coherence': 5.0, consistency': 5.0, fluency': 4.7)</p>

Figure E.2: SummEval Case Study. A summarization task where the initial response lacks cohesion and informativeness. Subsequent agents improve sentence structure, factual completeness, and coherence. Evaluations at each stage are conducted by Qwen2.5-7B-Instruct (frozen evaluator). The final summary exhibits significantly enhanced quality as judged by the evaluator, confirming the utility of MAC-SPGG in generation tasks.

E.1 Evaluation Details

HumanEval. To assess agents' code generation capabilities, we evaluate all models on the full HumanEval benchmark. Following standard practice, we adopt the *pass@1* metric—indicating the percentage of problems correctly solved by the first generated solution—as our main performance indicator.

MMLU. To evaluate MMLU, we measured the accuracy with which models were able to select the correct multiple-choice answer in each problem. We evaluated models on one hundred randomly selected MMLU questions randomly distributed across each of the subject areas.

GSM8K. To assess mathematical reasoning ability, we evaluate models on the GSM8K dataset, which consists of grade-school math word problems requiring multi-step arithmetic and logical reasoning. We randomly selected one hundred problems from the dataset and report accuracy as the percentage of problems for which the model produces the correct final numerical answer.

SummEval. To evaluate agents' natural language processing ability, we use models to test all the SummEval problems and also the 1600 examples and corresponding scores given by datasets, we used them to fine-tune our evaluator.

CNN Dailymail. We also used the datasets from Huggingface, which is similar to the SummEval, which contains 287,113 in its training subset. We used the 1.0.0 version to train our MAC-SPGG models.