BILEVEL INCENTIVE LEARNING FOR LANGUAGE MODEL SELF-IMPROVEMENT WITHOUT HUMAN LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-improvement enables large language models to learn from self-generated problems and automated feedback, but its effectiveness depends on the incentives governing problem generation and solution learning. We propose a bilevel post-training paradigm that jointly adapts these incentives and the policies they induce without human-provided solution labels. At the lower level, a questioner and a solver interact in a general-sum game under role-specific rewards. At the upper level, a shared incentive model minimizes the equilibrium solver's expected task failure. We propose BiGAP, a bilevel game-aware penalty method that constructs policy-specific objectives from Nikaido–Isoda gap functions and aggregates both players' gaps for incentive learning. Its training protocol coordinates question generation, separate practice and target-response sampling, policy optimization, and incentive adaptation, with optional target-question refresh. For fixed distributions and a fixed penalty parameter, we show under suitable regularity assumptions that, after T exact-oracle iterations, the average squared operator residual is $\mathcal{O}(T^{-1})$. The analysis further relates optimization accuracy and penalty strength to approximate Nash equilibrium guarantees. Across nine mathematical reasoning and formal theorem-proving benchmarks, BiGAP consistently outperforms fixed-incentive self-improvement. These gains are achieved without human-provided solution labels during self-improvement.

1 INTRODUCTION

Self-improvement offers a scalable post-training paradigm for large language models (LLMs), allowing them to learn from their own generations and task feedback. This is particularly attractive when human-written solutions are costly or unavailable: knowledge acquired during pretraining can support the generation of further learning experiences. Early approaches, including self-taught reasoner (STaR) and reinforced self-training with expectation-maximization (ReST$^{\mathrm{EM}}$), reuse successful generated solutions for training (Zelikman et al., 2022; Singh et al., 2023). Absolute Zero and R-Zero extend this process to problem generation without externally supplied training examples (Zhao et al., 2025; Huang et al., 2026a). The self-play theorem prover (STP) applies iterative conjecturing and proof verification to formal mathematics (Dong & Ma, 2025). These developments demonstrate that models can generate useful learning experiences from their existing reasoning capabilities.

Autonomous generation, however, does not ensure progress on the intended tasks. R-Zero reports declining estimated pseudo-label accuracy as generated questions become harder (Huang et al., 2026a). Its successor, R-Few, introduces human-labeled examples to guide question generation and jointly train the solver on human and synthetic data (Yu et al., 2025). Such guidance helps direct learning, but obtaining reliable solution labels can remain costly. The challenge is therefore to preserve autonomous practice generation while directing self-improvement toward the reasoning capabilities required by target tasks.

From self-play to incentive design. For a self-directed learner, an exercise is valuable when learning from it helps solve the problems that matter. In LLM self-play, rewards shape both the exercises generated and the responses reinforced. The self-guided self-play (SGS) method illustrates this principle through a frozen guide that discourages artificially complex conjectures and rewards relevance and clarity (Bailey et al., 2026). Yet the usefulness of an incentive depends on two

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

interacting learners: the questioner changes the solver's training distribution, and the solver changes the feedback available to the questioner. Their distinct objectives form a general-sum game. A Nash equilibrium (NE) rules out beneficial unilateral policy changes under given incentives, but does not establish target-task competence. Incentive design must therefore account for both learners' responses and their consequences for the solver's target-task performance.

A bilevel paradigm for self-improvement. We propose a post-training paradigm that learns the incentives governing question generation and solution learning. At the lower level (LL), a questioner and a solver optimize distinct rewards supplied by a shared incentive model. At the upper level (UL), this model is chosen to minimize the equilibrium solver's expected task failure. The two levels play complementary roles: learned rewards guide practice, while a fixed acceptance rule defines task success independently of the learned scores. Automatic feedback uses generated reference answers for mathematical questions and Lean verification for formal proofs, supporting task-directed learning without human-provided solution labels.

This design connects LLM self-improvement with bilevel learning (Franceschi et al., 2018), incentive adaptation, and optimization over game equilibria (Liu et al., 2022; Yang et al., 2022; Niu et al., 2025). Penalty methods provide related approaches to bilevel reinforcement learning (RL) and zero-sum LL games (Shen et al., 2025; Zheng et al., 2026). It places both language-model policies inside a general-sum equilibrium constraint, capturing how incentives shape the practice distribution and the policy trained on it. Section 2 formalizes the objective for fixed distributions; Section 3 gives a training instantiation with sampled interactions, evolving policies, and adaptive target-question selection.

The optimization variables distinguish this paradigm from existing LLM hierarchies. In evolving via asymmetric self-play (eva), prompts adapt under a fixed reward oracle (Ye et al., 2025). The self-optimization via asymmetric RL (SOAR) method rewards teachers through measured student improvement (Sundaram et al., 2026), while influence-guided self-evolution improves reasoning (INFUSER) rewards generators through optimizer-aware influence estimates (Chen et al., 2026). SOAR and INFUSER use labeled target problems to evaluate progress or estimate influence. Self-Guide also co-adapts policies and internal rewards for language agents (Wang et al., 2026). Our bilevel framework learns shared incentives above the coupled questioner–solver game, linking reward adaptation to target-task performance at the induced game equilibrium.

Coordinating the post-training process. Realizing this paradigm requires learning objectives and sampling procedures for three interacting models. We propose a bilevel game-aware penalty method (BiGAP) that constructs role-specific objectives using Nikaido–Isoda (NI) gaps. Each gap measures a player's potential improvement against its current opponent, and their sum vanishes exactly at an NE. Each policy objective penalizes its own gap, while the incentive objective incorporates both gaps. The resulting mixed operator need not be the gradient of a shared objective. This construction provides a common optimization structure for coordinating incentive learning with both policy updates.

The training protocol couples these objectives to two response streams: generated questions supply practice under learned rewards, and target questions supply direct task feedback to the solver. Collected interactions support policy training and incentive adaptation through sampled reward-gap surrogates. Updated incentives shape subsequent question and response generation, while optional target-question refresh changes the curriculum between iterations. This organization jointly adapts the data distribution, the policies, and their learned rewards through a repeated cycle of interaction collection, feedback construction, and training.

The major contributions of this work are highlighted as follows:

- **A bilevel post-training paradigm for self-improvement.** Our bilevel paradigm makes the incentives governing question generation and solution learning jointly learnable, and evaluates them through a general-sum LL equilibrium. The formulation connects autonomous practice generation to a target-task criterion that remains independent of learned reward scores, enabling reward learning without human-provided solution labels.
- **A game-aware training protocol with coupled data and model updates.** BiGAP couples player-specific NI penalties with an aggregate incentive objective. The protocol coordinates practice generation, independent LL and UL response sampling, policy optimization, and sampled incentive adaptation, jointly adapting the training data, rewards, and model parameters.
- **Convergence and equilibrium-accuracy guarantees.** For fixed distributions and a fixed penalty parameter, we show that T iterations of exact-oracle BiGAP yield an $\mathcal{O}(T^{-1})$ average squared operator-residual bound under Lipschitz and weak Minty regularity. Un-

der playerwise gradient dominance, the analysis further bounds LL Nash error in terms of the optimization residual and the chosen penalty strength.

- **Consistent gains across reasoning tasks and evaluation budgets.** Experiments cover four mathematical question-answering (QA) benchmarks and five theorem-proving benchmarks with Lean-verified proofs. Self-improvement uses no human-provided solution labels, and QA starts without external training examples. BiGAP can exceed fixed-incentive training across benchmarks, with higher average performance in both task domains under every reported evaluation budget.

Further comparisons with related methods appear in Appendix A.

2 Bilevel Incentive Design for Recursive Self-Improvement

In this work, we formulate incentive design for language model self-improvement as a bilevel optimization problem. The LL comprises a general-sum game between a questioner and a solver, whose policies are optimized under role-specific rewards parameterized by a shared incentive model. The UL optimizes this model to minimize the expected task failure of the solver at an induced Nash equilibrium, thereby aligning the learned incentives with downstream performance. We develop this formulation under fixed prompt and target-question distributions; practical data collection and adaptive updates to the target-question dataset are described in Section 3.

2.1 **Lower Level**: The Incentivized Questioner–Solver Game

The questioner policy π_ϕ and solver policy π_ψ are parameterized by $\phi \in \mathbb{R}^{d_\phi}$ and $\psi \in \mathbb{R}^{d_\psi}$, respectively. Their interaction consists of generating a practice question q from a prompt s and producing an answer a to q:

$$s \sim \nu_{\text{seed}}, \qquad q \sim \pi_\phi(\cdot \mid s), \qquad a \sim \pi_\psi(\cdot \mid q),$$

where ν_{seed} is a fixed sampling distribution over question-generation prompts. The task interface specifies admissible question and answer formats and a fixed acceptance rule $v(q, a) \in \{0, 1\}$, with value one for accepted responses and zero for invalid or rejected responses.

The questioner and solver receive role-specific rewards $r_{x,Q}(s, q, a)$ and $r_{x,S}(s, q, a)$, respectively, whose dependence on $x \in \mathbb{R}^{d_x}$ is determined by a shared incentive model. The questioner reward encourages useful practice, while the solver reward is nonzero only for accepted solutions; their constructions are detailed in Section 3.2. The resulting Kullback–Leibler (KL)-regularized expected rewards are

$$J_Q(x, \phi, \psi) \triangleq \mathbb{E}_{s,q,a}\left[r_{x,Q}(s, q, a) - \tau_Q \log \frac{\pi_\phi(q \mid s)}{\pi_Q^{\text{ref}}(q \mid s)} \right], \tag{1a}$$

$$J_S(x, \phi, \psi) \triangleq \mathbb{E}_{s,q,a}\left[r_{x,S}(s, q, a) - \tau_S \log \frac{\pi_\psi(a \mid q)}{\pi_S^{\text{ref}}(a \mid q)} \right], \tag{1b}$$

where π_Q^{ref} and π_S^{ref} are fixed reference policies and $\tau_Q, \tau_S > 0$ control the respective regularization strengths. Both expectations follow the interaction above and include any additional samples used to compute reward statistics. The corresponding LL losses are $g_Q \triangleq -J_Q$ and $g_S \triangleq -J_S$. The questioner's reward depends on solver outcomes, while the solver's training distribution depends on the questioner. These coupled objectives are not constrained to sum to zero and therefore define a general-sum game.

For fixed x, denote the Nash equilibrium set by $\mathcal{N}(x)$. A policy parameter pair $(\phi^\star, \psi^\star)$ belongs to $\mathcal{N}(x)$ precisely when neither player can reduce its own loss through a unilateral policy change:

$$\phi^\star \in \arg\min_\phi g_Q(x, \phi, \psi^\star), \qquad \psi^\star \in \arg\min_\psi g_S(x, \phi^\star, \psi). \tag{2}$$

The LL equilibrium characterizes policy responses to given incentives, while the UL objective guides incentive selection according to the resulting solver's target-task performance.

2.2 **Upper Level**: Task-Guided Incentive Design

An equilibrium of the LL game ensures mutual optimality under the specified rewards, but does not ensure that the resulting solver performs well on the target tasks. The UL therefore evaluates incentives through the target performance of the solver they induce. By placing the shared incentive model above both learners, this hierarchy accounts for how reward design shapes question generation and solution learning jointly. We construct the UL objective as the expected failure under the task acceptance rule:

$$f(x, \phi, \psi) \triangleq \mathbb{E}_{q \sim \mathcal{D}_{\text{UL}},\, a \sim \pi_\psi(\cdot \mid q)}[1 - v(q, a)]\,. \tag{3}$$

where $\mathcal{D}_{\mathrm{UL}}$ is a fixed, nonempty dataset of target questions and $q \sim \mathcal{D}_{\mathrm{UL}}$ denotes uniform sampling from it. The fixed acceptance rule v provides a task criterion independent of the learned reward scores, so increasing those scores alone cannot improve the UL objective.

Although f depends explicitly only on ψ, changing the incentives affects its value through the coupled equilibrium response of the questioner and solver. We thus optimize the incentive parameters and equilibrium policies jointly:

$$\min_{x,\phi,\psi} \quad f(x,\phi,\psi) \qquad \text{s.t.} \quad (\phi,\psi) \in \mathcal{N}(x). \tag{4}$$

This formulation selects rewards according to the target performance achievable at an induced LL equilibrium. When multiple equilibria exist, joint minimization implements optimistic equilibrium selection, favoring an equilibrium with the smallest UL loss; it does not assume that practical training necessarily reaches that equilibrium.

Acceptance feedback can be obtained without human-provided solution labels through formal proof verification or comparison with automatically generated reference answers fixed before response sampling. In the latter case, f measures reference-based acceptance failure, which may differ from true task error. Target questions may come from unlabeled collections or LL generation; Section 3 summarizes feedback collection and adaptive updates to $\mathcal{D}_{\mathrm{UL}}$, with task interfaces in Appendix B.7.

3 Incentive Learning and Policy Coevolution

Building on Section 2, BiGAP jointly learns incentives and policies through four stages: rollout collection, role-specific reward computation, policy and incentive optimization, and optional UL dataset evolution. Algorithm 1 gives the training loop; Appendix B provides implementation details.

3.1 Question Generation and UL Feedback

At iteration t, the prompt batch $\mathcal{U}_t$ mixes two modes under ν_{seed}. *Similarity* prompts use questions from $\mathcal{D}_{\mathrm{UL}}$ as seeds and request related, easier problems; *exploration* prompts request new problems without a seed. The questioner π_{ϕ_t} generates N_Q questions per prompt to form $\mathcal{Q}_t$. The task interface checks admissibility and fixes any required reference answers before solver responses are sampled. Only admissible questions reach the solver, while rejected proposals remain in the questioner's feedback. Appendix B.1 specifies prompt construction and admissibility.

For each admissible $q \in \mathcal{Q}_t$, the solver π_{ψ_t} samples $N_S > 2$ responses, collected in $\mathcal{A}(q)$. Their mean acceptance under $v(q,a)$ gives the empirical pass rate $\widehat{\rho}_t(q)$. Independently, it samples N_{UL} responses per UL question, yielding pass rates $\widehat{\rho}_t^{\mathrm{UL}}(q)$ and, for a nonempty dataset, the mean failure estimate $\widehat{f}_t$ of equation 3. Invalid responses count as failures. Both streams use the pre-update solver, and the dataset remains fixed throughout collection and model updates. Appendix B.2 gives the estimators, and Appendix B.7 gives the task-specific acceptance procedures.

3.2 Learnable Role-Specific Rewards

A shared LLM evaluator with parameters x assigns role-specific scores $h_{x,i}(s,q,a) \in [0,1]$ for $i \in \{Q,S\}$. Suppressing the common interaction arguments, we instantiate the rewards in equation 1 as

$$r_{x,i} = G_i P_i \left[c_{i,\mathrm{cls}} h_{x,i} + \sum_{k \in \mathcal{K}_i} c_{i,k} r_{i,k} \right]. \tag{5}$$

The binary gate G_Q requires an admissible question, while G_S additionally requires an accepted response. The factors $P_i \in [0,1]$ penalize excessive output length. All coefficients, gates, length rules, and auxiliary reward rules are fixed; x is the learnable component shared by both roles.

Questioner incentives. The auxiliary components $\mathcal{K}_Q = \{\mathrm{sim}, \mathrm{div}, \mathrm{diff}\}$ promote relevant, varied practice at a useful difficulty. Similarity rewards meaningful seed variants while discouraging exact repetition. For exploration, diversity combines novelty relative to the UL dataset with rarity among admissible exploration questions in the batch. Both modes receive a difficulty reward favoring $\widehat{\rho}_t(q)$ near a target $\tau_{\mathrm{diff}} \in (0,1)$; it vanishes when all sampled responses fail or all succeed. The action reward $R_{x,Q}(s,q)$ averages interaction rewards over $\mathcal{A}(q)$, with zero reward for inadmissible questions.

Solver incentives. For $\mathcal{K}_S = \varnothing$, each generated-question response receives $R_{x,S}(q,a) = r_{x,S}(s(q),q,a)$, where $s(q)$ is its originating prompt. Among accepted responses, the learned score supplies graded feedback; the UL task criterion remains the fixed acceptance rule, independent of that score. Appendix B.3 specifies all reward components and aggregation rules.

3.3 Policy Learning and Incentive Adaptation

We address equation 4 using a penalty method that balances target-task performance with separate unilateral optimality gaps for the two players (Shen et al., 2025; Zheng et al., 2026).

Best responses and equilibrium feasibility. Write $z = (x, \phi, \psi) \in \mathbb{R}^{d_z}$, where $d_z = d_x + d_\phi + d_\psi$. For fixed incentives and opponent policy, a *best response* globally minimizes the player's own LL loss. Assuming these minima are attained, choose $\phi^{\mathrm{br}} \in \arg\min_{\phi'} g_Q(x, \phi', \psi)$ and $\psi^{\mathrm{br}} \in \arg\min_{\psi'} g_S(x, \phi, \psi')$. The corresponding optimal values are $g_Q^\star(x, \psi) \triangleq g_Q(x, \phi^{\mathrm{br}}, \psi)$ and $g_S^\star(x, \phi) \triangleq g_S(x, \phi, \psi^{\mathrm{br}})$. Each player's maximum loss reduction through a unilateral policy change is therefore

$$\delta_Q(z) \triangleq g_Q(z) - g_Q^\star(x, \psi), \qquad \delta_S(z) \triangleq g_S(z) - g_S^\star(x, \phi). \tag{6}$$

Their sum $\delta(z) \triangleq \delta_Q(z) + \delta_S(z)$ is the *Nikaido–Isoda (NI) gap* (Nikaidô & Isoda, 1955; Raghunathan et al., 2019). Each term is nonnegative because a player can retain its current policy. Thus, $\delta(z) = 0$ exactly when $(\phi, \psi) \in \mathcal{N}(x)$, and $\delta(z)$ bounds either player's gain from unilateral deviation.

Penalized objectives and the mixed operator. For $\lambda > 0$, define the aggregate penalty objective $\mathcal{L}_\lambda(z) \triangleq f(z) + \lambda\delta(z)$ and the player-specific objectives $\mathcal{L}_{\lambda,i}(z) \triangleq f(z) + \lambda\delta_i(z)$ for $i \in \{Q, S\}$. With other blocks fixed, x follows $\mathcal{L}_\lambda$, while ϕ and ψ follow $\mathcal{L}_{\lambda,Q}$ and $\mathcal{L}_{\lambda,S}$, respectively, preserving the LL game's unilateral optimization structure.

The corresponding block gradients define the mixed first-order operator $\mathcal{H}_\lambda : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$:

$$\mathcal{H}_\lambda(z) \triangleq \begin{bmatrix} \nabla_x \mathcal{L}_\lambda(z) \\ \nabla_\phi \mathcal{L}_{\lambda,Q}(z) \\ \nabla_\psi \mathcal{L}_{\lambda,S}(z) \end{bmatrix} = \begin{bmatrix} \nabla_x f(z) + \lambda[\nabla_x \delta_Q(z) + \nabla_x \delta_S(z)] \\ \nabla_\phi f(z) + \lambda \nabla_\phi g_Q(z) \\ \nabla_\psi f(z) + \lambda \nabla_\psi g_S(z) \end{bmatrix}. \tag{7}$$

The policy blocks simplify because $g_Q^\star(x, \psi)$ is independent of ϕ, and $g_S^\star(x, \phi)$ is independent of ψ. Each policy block omits derivatives of the opponent's gap, so $\mathcal{H}_\lambda$ is generally a game operator rather than the gradient of a common scalar objective.

With the target distribution and policies fixed, f in equation 3 is independent of x. Thus, $\nabla_x f(z) = 0$, and incentive learning follows the aggregate gap gradient. Under the differentiable-envelope conditions in Section 4,

$$\nabla_x \delta_Q(z) = \nabla_x g_Q(z) - \nabla_x g_Q(x, \phi^{\mathrm{br}}, \psi), \quad \nabla_x \delta_S(z) = \nabla_x g_S(z) - \nabla_x g_S(x, \phi, \psi^{\mathrm{br}}). \tag{8}$$

These envelope gradients require no differentiation through best responses. The solver receives task guidance through $\nabla_\psi f(z)$; its evolving policy transmits this feedback to incentive learning, which aims to reduce both players' optimality gaps.

Policy optimization. Because $g_i = -J_i$, the policy blocks minimize $f - \lambda J_i$. With f independent of ϕ, the questioner maximizes its regularized reward; the solver combines generated-practice rewards with UL task loss $1 - v(q, a)$. We implement both updates using group relative policy optimization (GRPO) surrogates (Shao et al., 2024) with $x = x_t$ and recorded rewards fixed. The solver's LL and UL losses are averaged separately and combined. Appendix B.4 gives the losses and reference regularization.

Incentive optimization. We approximate the best-response comparisons using the same stored interactions. For role i, let $\mathcal{C}_i$ contain its contexts and $\mathcal{Y}_i(c)$ the nonempty candidate collection under context c: generation prompts and their questions for Q, admissible generated questions and their responses for S. The empirical reward gap is

$$\widehat{\delta}_i(x) \triangleq \frac{1}{|\mathcal{C}_i|} \sum_{c \in \mathcal{C}_i} \left[\max_{y \in \mathcal{Y}_i(c)} R_{x,i}(c, y) - \frac{1}{|\mathcal{Y}_i(c)|} \sum_{y \in \mathcal{Y}_i(c)} R_{x,i}(c, y) \right]. \tag{9}$$

An absent role contributes zero. For nonempty $\mathcal{D}_{\mathrm{UL}}$, the incentive model minimizes $\widehat{\mathcal{L}}_\lambda(x) = \widehat{f}_t + \lambda \sum_{i \in \{Q,S\}} \widehat{\delta}_i(x)$. The recorded $\widehat{f}_t$ is constant in x, so only the gap terms contribute gradients. Candidate outputs and acceptance outcomes remain fixed while learned scores are recomputed; no gradient passes through the policy updates. These finite-candidate reward gaps omit the policy optimization and KL costs of an exact best response. Appendix B.5 gives the implementation, while Section 4 analyzes anchored updates with exact best-response and gradient oracles under fixed distributions.

Algorithm 1 BiGAP: A Bilevel Game-Aware Penalty Method

Require: Initial parameters (x_0, ϕ_0, ψ_0); UL dataset $\mathcal{D}_{\mathrm{UL}}$ (possibly empty); sample counts $N_Q, N_S, N_{\mathrm{UL}}$; penalty parameter $\lambda > 0$; iterations T.
for $t = 0, \ldots, T-1$ **do**
Phase 1: Interaction collection and task feedback
Form the prompt batch $\mathcal{U}_t$ using similarity and exploration prompts; use exploration only if $\mathcal{D}_{\mathrm{UL}} = \varnothing$.
Generate N_Q questions per prompt with π_{ϕ_t} to form $\mathcal{Q}_t$.
Check question admissibility and prepare any required references before response sampling.
For each admissible $q \in \mathcal{Q}_t$, collect $\mathcal{A}(q) = (a_{t,q,j})_{j=1}^{N_S}$ under π_{ψ_t} and compute $\widehat{\rho}_t(q)$ by equation 14a.
if $\mathcal{D}_{\mathrm{UL}} \neq \varnothing$ **then**
Independently sample N_{UL} responses $a_{t,q,j}^{\mathrm{UL}}$ under π_{ψ_t} for each $q \in \mathcal{D}_{\mathrm{UL}}$.
Record $\widehat{f}_t$ and $\widehat{\rho}_t^{\mathrm{UL}}(q)$ using equation 16 and equation 14b.
end if
Phase 2: Role-reward computation
Compute $r_{x_t,Q}$ and $r_{x_t,S}$ by equation 5, using the empty-dataset novelty convention when needed.
Form question rewards $R_{x_t,Q}$ by equation 24 and response rewards $R_{x_t,S}(q,a) = r_{x_t,S}(s(q),q,a)$.
Phase 3: Policy and incentive optimization
Update $\phi_t \to \phi_{t+1}$ and $\psi_t \to \psi_{t+1}$ using the GRPO losses in equation 30, with $x = x_t$ and recorded rewards fixed; omit the UL loss when $\mathcal{D}_{\mathrm{UL}} = \varnothing$.
if $\mathcal{D}_{\mathrm{UL}} \neq \varnothing$ **then**
Form $\widehat{\delta}_i(x) = \widehat{R}_i^{\mathrm{best}}(x) - \widehat{R}_i(x)$, $i \in \{Q, S\}$, using equation 32.
Update $x_t \to x_{t+1}$ by optimizing $\widehat{\mathcal{L}}_\lambda(x)$ in equation 33 on the same stored candidates.
else
Set $x_{t+1} = x_t$.
end if
Phase 4: UL dataset refresh
if dataset updates are enabled **then**
Select $\mathcal{D}_{\mathrm{UL}}^{\mathrm{add}}$ from $\mathcal{Q}_t$ using the recorded $\widehat{\rho}_t(q)$ and the admission rule equation 34.
Form $\mathcal{D}_{\mathrm{UL}}^{\mathrm{keep}}$ using the recorded $\widehat{\rho}_t^{\mathrm{UL}}(q)$ and equation 35; set it to $\varnothing$ if $\mathcal{D}_{\mathrm{UL}} = \varnothing$.
Set $\mathcal{D}_{\mathrm{UL}} \leftarrow \mathcal{D}_{\mathrm{UL}}^{\mathrm{keep}} \cup \mathcal{D}_{\mathrm{UL}}^{\mathrm{add}}$.
end if
end for
return $(\pi_{\phi_T}, \pi_{\psi_T}, x_T, \mathcal{D}_{\mathrm{UL}})$.

3.4 Updating the UL Dataset

After the model updates, we optionally refresh $\mathcal{D}_{\mathrm{UL}}$ using feedback already collected under π_{ψ_t}, without reevaluating the updated solver. Admissible generated questions with positive but sufficiently low $\widehat{\rho}_t(q)$ are considered for admission after deduplication. Existing targets with high $\widehat{\rho}_t^{\mathrm{UL}}(q)$ can be retired. The retained and admitted questions provide both UL targets and generation seeds for the next iteration. This dataset evolution lies outside the fixed-distribution formulation and analysis; Appendix B.6 gives selection thresholds, budgets, and retention options.

4 Convergence Guarantees for BiGAP

We analyze an exact-oracle version of BiGAP, detailed in Appendix D. The mixed operator in equation 7 is generally not the gradient of a shared objective, so standard objective-descent arguments do not directly apply to BiGAP. Under Lipschitz continuity and a weak Minty condition, we establish an $\mathcal{O}(T^{-1})$ bound on the average squared operator residual. With playerwise gradient dominance, this bound further controls each player's potential gain from a unilateral deviation, separating finite-iteration error from finite-penalty error. These guarantees quantify how accurately BiGAP satisfies the LL equilibrium constraint. The analysis fixes the prompt distribution, UL dataset, and penalty parameter λ, and assumes exact best-response and gradient oracles.

4.1 Exact-Oracle Penalty Algorithm

Under Assumption D.2, the best-response value functions in equation 6 are differentiable. Exact best responses therefore allow the incentive block of $\mathcal{H}_\lambda$ to be evaluated through envelope gradients, without differentiating through the best-response policies. The policy blocks use gradients of their respective penalized objectives. Lemma D.6 establishes Lipschitz continuity of this operator.

To stabilize the coupled updates, introduce an anchor $w \in \mathbb{R}^{d_z}$ and define $\mathcal{F}_\lambda(z, w) \triangleq \mathcal{H}_\lambda(z) + \rho(z - w)$, where $\rho > 0$ controls the anchoring term. Algorithm 2 updates all three parameter

blocks simultaneously with step size η, then updates the anchor with relaxation parameter β. All best responses and gradients are evaluated at the current iterate $z_t = (x_t, \phi_t, \psi_t)$. The convergence parameters specified below satisfy $\eta\rho = 1$.

4.2 From Penalized Stationarity to LL Equilibrium

Stationarity of incentives and policies. Let L_H denote the Lipschitz constant of $\mathcal{H}_\lambda$ established in Lemma D.6. Assumption D.7 requires a zero $z^\star$ satisfying $\langle \mathcal{H}_\lambda(z), z - z^\star \rangle \geq -\kappa \|\mathcal{H}_\lambda(z)\|^2$ for all $z \in \mathbb{R}^{d_z}$, where $\kappa \geq 0$. The condition $\kappa L_H < 1$ controls the allowed negative alignment relative to the operator's Lipschitz constant and yields the following convergence guarantee.

Theorem 4.1 (Convergence of the exact-oracle algorithm)**.** *Under Assumptions D.2 and D.7, suppose $L_H > 0$ and $\kappa L_H < 1$. Apply Algorithm 2 with exact best-response and gradient oracles and choose*

$$\rho = \frac{2L_H}{1 + \kappa L_H}, \qquad \eta = \frac{1 + \kappa L_H}{2L_H}, \qquad \beta = \frac{(1 - \kappa L_H)^2}{16(1 + \kappa L_H)}.$$

Then, for every $T \geq 1$,

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\mathcal{H}_\lambda(z_t)\|^2 \leq \frac{128 L_H^2 \|z_0 - z^\star\|^2}{(1 - \kappa L_H)^3 T}.$$

The returned iterate $z_{\widehat{t}}$ therefore satisfies

$$\max\{\|\nabla_x \mathcal{L}_\lambda(z_{\widehat{t}})\|, \|\nabla_\phi \mathcal{L}_{\lambda,Q}(z_{\widehat{t}})\|, \|\nabla_\psi \mathcal{L}_{\lambda,S}(z_{\widehat{t}})\|\} \leq \|\mathcal{H}_\lambda(z_{\widehat{t}})\| = \mathcal{O}(T^{-1/2}).$$

Thus, the residual simultaneously controls stationarity of the incentive objective with respect to x and each policy objective with respect to its own parameters. At fixed λ, $\mathcal{O}(\epsilon^{-2})$ exact-oracle iterations suffice to make all three gradient norms at most ϵ.

From stationarity to equilibrium feasibility. To relate this guarantee to the LL constraint in equation 4, assume playerwise gradient dominance with constants $\mu_Q, \mu_S > 0$: each player's best-response gap is bounded by a constant multiple of its squared policy-gradient norm (Assumption D.9). Write $\mu = \min\{\mu_Q, \mu_S\}$, and let C_f denote the Lipschitz constant of f.

Corollary 4.2 (Approximate Nash equilibrium under gradient dominance)**.** *Under the conditions of Theorem 4.1, additionally suppose Assumption D.9 holds. Then the returned iterate satisfies $\delta_Q(z_{\widehat{t}}) + \delta_S(z_{\widehat{t}}) \leq \epsilon_{\mathrm{NE}}(T)$, where*

$$\epsilon_{\mathrm{NE}}(T) = \frac{1}{2\mu\lambda^2} \left(C_f + \frac{8\sqrt{2} L_H \|z_0 - z^\star\|}{(1 - \kappa L_H)^{3/2} \sqrt{T}} \right)^2.$$

Consequently, $(\phi_{\widehat{t}}, \psi_{\widehat{t}})$ is an $\epsilon_{\mathrm{NE}}(T)$-Nash equilibrium of the LL game at $x_{\widehat{t}}$.

This bounds either player's possible loss reduction through a unilateral policy change with incentives and the opponent fixed. At fixed λ, the bound approaches $C_f^2/(2\mu\lambda^2)$ as T increases. This finite-penalty term accounts for the UL task gradient: stationarity of the penalized policy objectives need not make the LL loss gradients vanish.

Penalty selection and oracle complexity. For $\lambda > 0$, the mixed operator has Lipschitz constant $L_H = \mathcal{O}(1 + \lambda)$ (Lemma D.6), while C_f is independent of λ. If $\|z_0 - z^\star\| = \mathcal{O}(1)$ and $1 - \kappa L_H \geq \varepsilon_0 > 0$ uniformly over the penalty values considered, then $\mathcal{O}((1 + \lambda)^2 \epsilon^{-2})$ exact-oracle iterations suffice to obtain $\|\mathcal{H}_\lambda(z)\| \leq \epsilon$. For $T \geq 1$, the Nash error bound separates into finite-penalty and finite-iteration terms:

$$\epsilon_{\mathrm{NE}}(T) \leq \frac{C_f^2}{\mu\lambda^2} + \frac{128 L_H^2 \|z_0 - z^\star\|^2}{\mu\lambda^2 (1 - \kappa L_H)^3 T} = \mathcal{O}\left(\frac{1}{\mu\lambda^2} + \frac{1}{\mu T} \right). \tag{10}$$

If μ is uniformly bounded away from zero, choosing $\lambda = \Theta(\varepsilon^{-1/2})$ and $T = \Theta(\varepsilon^{-1})$ with sufficiently large constants yields an ε-Nash equilibrium. To also ensure $\|\mathcal{H}_\lambda(z_{\widehat{t}})\| \leq \epsilon$ for $0 < \epsilon \leq 1$, the same penalty choice gives a joint guarantee in $\mathcal{O}(\varepsilon^{-1}\epsilon^{-2})$ exact-oracle iterations. These scalings require the stated uniform bounds, including preservation of the weak Minty margin as λ increases. Proofs appear in Appendix D.

5 Evaluating BiGAP for Recursive Self-Improvement in Reasoning

5.1 Experimental Setup

We evaluate mathematical QA with Qwen3-4B-Base on MATH-500 (Lightman et al., 2023; Hendrycks et al., 2021) and AIME24/25/26 (Maxwell-Jia, 2024; MathArena, 2025; 2026). For

Table 1: QA comparison with Qwen3-4B-Base solvers (%). Avg.$_3$ is the unweighted mean of the three reported benchmark scores. L: evaluation completion cap; N/A: cap not confirmed for the cited scores. Data tags describe self-improvement inputs: Z = no external training examples; C = external corpus; T = external task-generating teacher; D = labeled development feedback during training. Bold compares our configurations within each budget.

Method	Data	L	MATH-500	AIME24	AIME25	Avg.$_3$
R-Zero (Huang et al., 2026a)	Z	N/A	79.60	13.40	9.60	34.20
SPIRAL-Multi (Liu et al., 2026b)	Z	N/A	78.20	19.70	13.30	37.07
SPICE (Liu et al., 2025)	C	N/A	78.00	12.20	19.10	36.43
Strong Challenger (Liu et al., 2025)	C+T	N/A	78.00	12.70	12.90	34.53
INFUSER (Chen et al., 2026)	C+D	8,192	76.65	11.35	10.73	32.91
Initial (ours)	Z	4,096	60.98	9.58	10.62	27.06
Fixed reward ($t = 2$)	Z	4,096	64.29	12.50	11.46	29.42
BiGAP ($t = 4$)	Z	4,096	**65.76**	**13.54**	**12.92**	**30.74**
Initial (ours)	Z	8,192	68.66	37.08	26.25	44.00
Fixed reward ($t = 3$)	Z	8,192	74.86	38.54	28.33	47.24
BiGAP ($t = 2$)	Z	8,192	**75.16**	**39.79**	**28.96**	**47.97**

Table 2: Mathematical QA accuracy (%) at 4,096- and 8,192-token completion limits. Bold: the highest value; underlining marks: the second-highest distinct value. Relative gains compare each column's best BiGAP and best fixed-reward score.

Method	t	MATH-500		AIME24		AIME25		AIME26		Avg.$_4$	
		4,096	8,192	4,096	8,192	4,096	8,192	4,096	8,192	4,096	8,192
Initial	0	60.98	68.66	9.58	37.08	10.62	26.25	8.75	31.46	22.48	40.86
Fixed	1	63.21	74.67	9.17	38.12	9.79	26.04	8.33	33.12	22.63	42.99
BiGAP	1	63.95	75.06	11.67	38.33	10.21	28.12	9.17	33.26	23.75	43.69
Fixed	2	64.29	74.76	12.50	37.29	11.46	26.88	8.54	35.00	24.20	43.48
BiGAP	2	64.68	75.16	12.29	**39.79**	11.46	28.96	9.79	35.42	24.56	44.83
Fixed	3	64.14	74.86	10.00	38.54	11.67	28.33	10.00	33.96	23.95	43.92
BiGAP	3	65.28	75.01	13.12	39.38	12.08	**29.17**	9.24	**36.04**	24.93	**44.90**
Fixed	4	64.68	74.88	11.46	37.92	11.46	27.08	9.58	32.50	24.30	43.10
BiGAP	4	**65.76**	**75.29**	**13.54**	39.03	**12.92**	28.06	**10.21**	35.07	**25.61**	44.36
Fixed	5	64.36	74.97	10.69	38.33	10.83	27.64	8.12	33.33	23.50	43.57
BiGAP	5	65.11	74.86	11.94	38.75	11.88	29.10	9.17	34.38	24.53	44.27
Gain (%)		+1.67	+0.43	+8.32	+3.24	+10.71	+2.97	+2.10	+2.97	+5.40	+2.23

formal theorem proving, we evaluate Qwen3.5-4B on miniF2F (Zheng et al., 2022), ProofNet (Azerbayev et al., 2023), ProverBench (Ren et al., 2025), MathOlympiadBench (MOBench) (Lin et al., 2025), and CAM-Bench (Long et al., 2026). QA reports mean answer accuracy at 4,096/8,192-token completion limits; proving reports Lean-verified pass@4/pass@16. Evaluation uses 16 responses per question and reports the averaged results of 3 runs. Appendix C gives the details.

Fixed reward trains the solver with fixed LL rewards on generated questions and current seeds, refreshing seeds from their response pass rates. It omits the separate UL solver task loss and incentive updates; **BiGAP** includes both. Both tasks report five training iterations plus initialization. Absolute gains are in percentage points (pp); relative gains are $100(b - f)/f$ for BiGAP and fixed scores b and f.

5.2 Comparison with Existing Frameworks

External scores in Table 1 are reference points because evaluation completion caps are not confirmed for all cited results and answer-checking procedures differ. At 4,096 tokens, BiGAP's AIME scores are numerically comparable to several published results: its 13.54% on AIME24 exceeds four of the five external scores, while its 12.92% on AIME25 exceeds R-Zero and INFUSER. Its Avg.$_3$ exceeds fixed reward by 1.32 pp. However, MATH-500 accuracy remains below all listed external methods at both budgets.

5.3 Effectiveness Across Tasks

BiGAP achieves higher peak averages and usually leads at matched iterations. Across each method's best checkpoints at iterations 1–5, BiGAP gains 1.31/0.98 pp on QA Avg.$_4$ at 4,096/8,192

Table 3: Formal theorem-proving results (%) with pass@4 and pass@16. Avg.$_5$ is the unweighted mean over all five benchmarks, rounded after averaging. Bold: the highest value; underlining marks: the second-highest distinct value. Relative gains compare each column's best BiGAP and best fixed-reward score.

Method	t	miniF2F		ProofNet		ProverBench		MOBench		CAM-Bench		Avg.$_5$	
		pass@4	pass@16	pass@4	pass@16	pass@4	pass@16	pass@4	pass@16	pass@4	pass@16	pass@4	pass@16
Initial	0	27.86	36.34	3.33	4.58	13.21	18.97	0.59	1.11	0.79	1.97	9.16	12.59
Fixed	1	28.92	38.11	**3.86**	6.20	14.84	21.23	0.92	1.67	1.62	2.63	10.03	13.97
BiGAP	1	29.72	38.52	3.51	5.48	14.92	21.54	**1.05**	**1.94**	1.64	2.57	10.17	14.01
Fixed	2	30.24	38.80	3.73	5.57	14.86	22.46	0.91	1.85	1.57	2.30	10.26	14.20
BiGAP	2	29.83	38.93	3.37	5.12	14.90	22.15	0.74	1.39	1.62	2.43	10.09	14.00
Fixed	3	29.03	38.52	3.75	5.93	14.79	21.64	0.75	1.39	1.70	2.73	10.00	14.04
BiGAP	3	**30.93**	**40.16**	3.75	**6.74**	**15.01**	**23.08**	0.88	1.48	1.72	**2.83**	**10.46**	**14.86**
Fixed	4	29.14	38.93	3.41	5.57	14.00	20.92	0.82	1.20	**1.74**	2.57	9.82	13.84
BiGAP	4	30.30	39.34	3.62	5.57	14.62	22.77	0.86	1.39	1.58	2.70	10.20	14.35
Fixed	5	29.20	38.66	3.58	6.11	14.23	21.23	0.77	1.30	1.68	2.63	9.89	13.99
BiGAP	5	30.76	39.75	3.61	6.02	14.74	22.36	0.72	1.39	1.65	2.60	10.30	14.42
Gain (%)		+2.28	+3.16	-2.85	+8.71	+1.01	+2.76	+14.13	+4.86	-1.15	+3.66	+1.91	+4.66

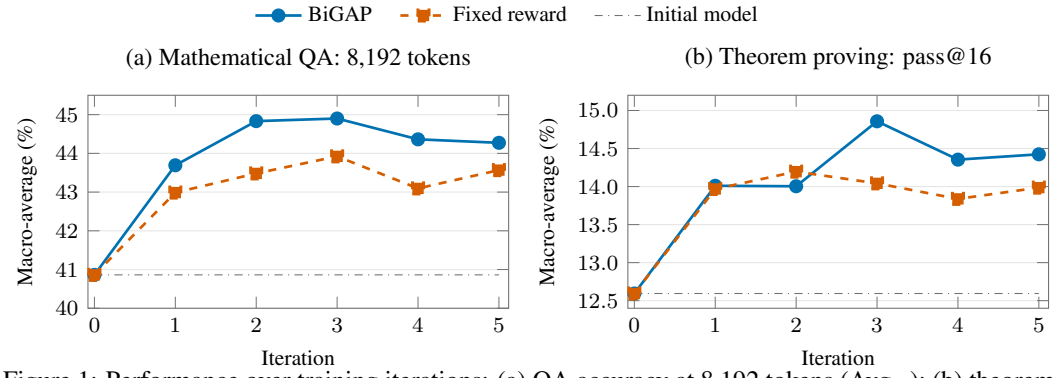


Figure 1: Performance over training iterations: (a) QA accuracy at 8,192 tokens (Avg.$_4$); (b) theorem-proving pass@16 (Avg.$_5$). Blue circles: BiGAP; dashed orange squares: fixed rewards; gray lines: initial solver. Values are from Tables 2 and 3. Panel scales differ; lines connect recorded checkpoints and do not imply uncertainty estimates or equal compute.

tokens and 0.20/0.66 pp on proving Avg.$_5$ at pass@4/pass@16 (Tables 2 and 3). Improvements over initialization are 3.13/4.04 pp for QA and 1.30/2.26 pp for proving. It leads on macro averages in all ten matched QA comparisons and eight of ten proving comparisons. QA peaks at $t = 4$/$t = 3$ for 4,096/8,192 tokens; both proving averages peak at $t = 3$. Table 1's three-benchmark mean instead selects $t = 2$ at 8,192 tokens. Fixed reward retains the best pass@4 scores on ProofNet and CAM-Bench. This comparison assesses the full bilevel procedure, including incentive adaptation and UL solver task feedback.

5.4 Scaling with Training Iterations

In Figure 1, BiGAP peaks at iteration 3: 44.90% for QA at 8,192 tokens and 14.86% for proving at pass@16. Scores decline slightly thereafter, but iteration 5 retains advantages of 0.71 pp and 0.44 pp, respectively, over fixed reward. This pattern may reflect overfitting to generated questions or accumulated errors in self-generated supervision.

6 Conclusion

We presented a bilevel game framework for LLM self-improvement that learns shared incentives through a coupled questioner-solver game. The framework connects incentive learning, question generation, and solver training through a practical procedure combining sampled reward gap updates and guided practice generation. We further provided an exact-oracle analysis that characterizes the penalty update structure, offering theoretical insight into the proposed algorithm. Experiments on mathematical QA and formal theorem proving show that the bilevel framework can outperform LL-only approaches.

AI USE STATEMENT

The authors developed the overall framework and mathematical formulation, wrote the proofs and initial manuscript, and independently implemented the core algorithm. We used generative AI tools to generate synthetic data, translate and polish the author-written draft, identify and summarize related work, and check the author-written proofs for potential errors. AI coding assistance focused on implementing distributed training and inference support and organizing the code repository. AI assistance with the theoretical content was limited to correctness checking. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies bilevel incentive learning for LLM self-improvement in mathematical reasoning and formal theorem proving, using existing benchmarks and model-generated practice questions. Self-generated data can contain errors, and reward optimization may exploit weaknesses in automated verification. We use task-specific answer checking and Lean proof validation, while recognizing the limitations of these checks. Improved reasoning capabilities may also support harmful applications. Our findings are limited to the evaluated tasks and do not establish the safety or reliability of broader deployment.

REPRODUCIBILITY STATEMENT

The training procedure is specified in Section 3 and Algorithm 1, with detailed objectives, reward construction, update rules, and task interfaces in Appendix B. We also provide our code implementation in an anonymous GitHub repository (url) for review to facilitate reproducibility and will make the code publicly available upon publication.

REFERENCES

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023. URL https://arxiv.org/abs/2302.12433.

Luke Bailey, Kaiyue Wen, Kefan Dong, Tatsunori Hashimoto, and Tengyu Ma. Scaling self-play with self-guidance. *arXiv preprint arXiv:2604.20209*, 2026. URL https://arxiv.org/abs/2604.20209.

Lesi Chen, Jing Xu, and Jingzhao Zhang. On finding small hyper-gradients in bilevel optimization: Hardness results and improved analysis. In *Conference on Learning Theory (COLT)*, volume 247, pp. 947–980, 2024a.

Lesi Chen, Yaohua Ma, and Jingzhao Zhang. Near-optimal nonconvex-strongly-convex bilevel optimization with fully first-order oracles. *Journal of Machine Learning Research*, 26(109):1–56, 2025.

Siyu Chen, Miao Lu, Beining Wu, Heejune Sheen, Fengzhuo Zhang, Shuangning Li, Zhiyuan Li, Jose Blanchet, Tianhao Wang, and Zhuoran Yang. INFUSER: Influence-guided self-evolution improves reasoning. *arXiv preprint arXiv:2606.09052v4*, 2026. URL https://arxiv.org/abs/2606.09052v4.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *International Conference on Machine Learning (ICML)*, 2024b.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026. URL https://arxiv.org/abs/2605.00674.

Kefan Dong and Tengyu Ma. STP: Self-play LLM theorem provers with iterative conjecturing and proving. In *International Conference on Machine Learning (ICML)*, 2025.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning (ICML)*, pp. 1568–1577, 2018.

Saeed Ghadimi and Mengdi Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018. URL https://arxiv.org/abs/1802.02246.

Panagiotis D. Grontas, Giuseppe Belgioioso, Carlo Cenedese, Marta Fochesato, John Lygeros, and Florian Dörfler. BIG Hype: Best intervention in games via distributed hypergradient descent. *IEEE Transactions on Automatic Control*, 69(12):8338–8353, 2024. doi: 10.1109/TAC.2024.3410890.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-Zero: Self-evolving reasoning LLM from zero data. In *International Conference on Learning Representations (ICLR)*, volume 2026, 2026a.

Siyuan Huang, Pengyu Cheng, Haotian Liu, Tao Chen, Yihao Liu, Jingwei Ni, Shijie Zhou, Ziyi Yang, Gangwei Jiang, Mengyu Zhou, Yu Cheng, Xiaoxi Jiang, and Guanjun Jiang. Skill self-play: Pushing the frontier of LLM capability with co-evolving skills. *arXiv preprint arXiv:2607.22529*, 2026b. URL https://arxiv.org/abs/2607.22529.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 4882–4892, 2021.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, JD Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D. Nowak. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning (ICML)*, 2023.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert Nowak. On penalty methods for non-convex bilevel optimization and first-order stochastic approximation. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiayang Li, Jing Yu, Boyi Liu, Yu Nie, and Zhaoran Wang. Achieving hierarchy-free approximation for bilevel programs with equilibrium constraints. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 20312–20335, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Yong Lin, Shange Tang, Bohan Lyu, Ziran Yang, Jui-Hui Chung, Haoyu Zhao, Lai Jiang, Yihan Geng, Jiawei Ge, Jingruo Sun, Jiayun Wu, Jiri Gesi, Ximing Lu, David Acuna, Kaiyu Yang, Hongzhou Lin, Yejin Choi, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-Prover-V2: Scaling formal theorem proving with scaffolded data synthesis and self-correction. *arXiv preprint arXiv:2508.03613*, 2025. URL https://arxiv.org/abs/2508.03613.

Zi Lin, Sheng Shen, Ilia Kulikov, Jingbo Shang, Jason Weston, and Yixin Nie. Learning to solve and verify: A self-play framework for code and test generation. *arXiv preprint arXiv:2502.14948v4*, 2026. URL https://arxiv.org/abs/2502.14948v4.

Bo Liu, Chuanyang Jin, Seungone Kim, Weizhe Yuan, Wenting Zhao, Ilia Kulikov, Xian Li, Sainbayar Sukhbaatar, Jack Lanchantin, and Jason Weston. SPICE: Self-play in corpus environments improves reasoning. *arXiv preprint arXiv:2510.24684*, 2025. URL https://arxiv.org/abs/2510.24684.

Bo Liu, Simon Yu, Yiding Jiang, Ao Qu, Andrew Zhao, Zichen Liu, Junsu Kim, Zijian Zhou, Seungone Kim, Tongzheng Ren, Mickel Liu, Hanfei Yu, Zhaorun Chen, Weiyan Shi, Paul Pu Liang, Luke Zettlemoyer, Yejin Choi, and Natasha Jaques. SPADE: Self-play in adaptive synthetic executable environments. *arXiv preprint arXiv:2608.19197*, 2026a. URL https://arxiv.org/abs/2608.19197v3.

Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 35407–35434, 2026b.

Boyi Liu, Jiayang Li, Zhuoran Yang, Hoi-To Wai, Mingyi Hong, Yu Nie, and Zhaoran Wang. Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 29001–29013, 2022.

Wentao Long, Yunfei Zhang, Chenyi Li, Li Zhou, Chumin Sun, and Zaiwen Wen. CAM-Bench: A benchmark for computational and applied mathematics in Lean. *arXiv preprint arXiv:2605.17255*, 2026. URL https://arxiv.org/abs/2605.17255.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108, pp. 1540–1552, 2020.

Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press, 1996. doi: 10.1017/CBO9780511983658. URL https://doi.org/10.1017/CBO9780511983658.

MathArena. AIME 2025. Hugging Face dataset, 2025. URL https://huggingface.co/datasets/MathArena/aime_2025. Public release of the Mathematical Association of America's 2025 AIME problems. Accessed September 25, 2026.

MathArena. AIME 2026. Hugging Face dataset, 2026. URL https://huggingface.co/datasets/MathArena/aime_2026. Public release of the Mathematical Association of America's 2026 AIME problems. Accessed September 25, 2026.

Maxwell-Jia. AIME 2024 Dataset. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/Maxwell-Jia/AIME_2024. Public release of the Mathematical Association of America's 2024 AIME I and II problems. Accessed September 25, 2026.

Hukukane Nikaidô and Kazuo Isoda. Note on noncooperative convex games. *Pacific Journal of Mathematics*, 5:807–815, 1955. doi: 10.2140/pjm.1955.5.807.

April Niu, Kai Wang, and Juba Ziani. Finding a multiple follower Stackelberg equilibrium: A fully first-order method. *arXiv preprint arXiv:2509.08161*, 2025. URL https://arxiv.org/abs/2509.08161.

Arvind Raghunathan, Anoop Cherian, and Devesh Jha. Game theoretic optimization via gradient-based Nikaido–Isoda function. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 5291–5300, 2019.

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*, 2025. URL https://arxiv.org/abs/2504.21801.

Sepideh Samadi and Farzad Yousefian. Improved guarantees for optimal Nash equilibrium seeking and bilevel variational inequalities. *SIAM Journal on Optimization*, 35(1):369–399, 2025. doi: 10.1137/23M1589402.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 30992–31015, 2023.

Han Shen, Zhuoran Yang, and Tianyi Chen. Principled penalty-based methods for bilevel reinforcement learning and RLHF. *Journal of Machine Learning Research*, 26(114):1–49, 2025.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023. URL https://arxiv.org/abs/2312.06585v4.

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching models to teach themselves: Reasoning at the edge of learnability. In *International Conference on Machine Learning (ICML)*, 2026.

Xinyu Wang, Hanwei Wu, Jingwei Song, Shuyuan Zhang, Jiayi Zhang, Fanqi Kong, Tung Sum Thomas Kwok, Xiao-Wen Chang, Yuyu Luo, Chenglin Wu, and Bang Liu. Co-evolution of policy and internal reward for language agents. *arXiv preprint arXiv:2604.03098*, 2026. URL https://arxiv.org/abs/2604.03098.

Jiachen Yang, Ethan Wang, Rakshit Trivedi, Tuo Zhao, and Hongyuan Zha. Adaptive incentive design with multi-agent meta-gradient reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 1436–1445, 2022.

Ziyu Ye, Rishabh Agarwal, Tianqi Liu, Rishabh Joshi, Sarmishta Velury, Quoc V. Le, Qijun Tan, and Yuan Liu. Reward-guided prompt evolving in reinforcement learning for LLMs. In *International Conference on Machine Learning (ICML)*, 2025.

Wenhao Yu, Zhenwen Liang, Chengsong Huang, Kishan Panaganti, Tianqing Fang, Haitao Mi, and Dong Yu. Guided self-evolving LLMs with minimal human supervision. *arXiv preprint arXiv:2512.02472*, 2025. URL https://arxiv.org/abs/2512.02472.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. Self-rewarding language models. In *International Conference on Machine Learning (ICML)*, pp. 57905–57923, 2024.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022.

Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. Quiet-STaR: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 64735–64772, 2024.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 105816–105879, 2025.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: A cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*, 2022.

Zihao Zheng, Irwin King, and Songtao Lu. Bilevel optimization over saddle points of zero-sum markov games. In *International Conference on Machine Learning (ICML)*, 2026.

A Related Work

Self-improvement for LLMs. Recent LLM self-improvement increasingly learns the practice distribution alongside the solver. Absolute Zero and R-Zero co-evolve problem generation and solving (Zhao et al., 2025; Huang et al., 2026a), while SPIRAL develops reasoning through multi-turn zero-sum games (Liu et al., 2026b). SPICE grounds generated questions in a corpus (Liu et al., 2025), and R-Few guides the evolving curriculum with a small set of human-provided anchors (Yu et al., 2025). In formal theorem proving, STP alternates conjecturing and proving (Dong & Ma, 2025), and SGS introduces a Guide, frozen during self-play, to score conjectures for target relevance and clarity (Bailey et al., 2026). Skill-SP uses execution feedback to adapt a library of task-generation skills and validators (Huang et al., 2026b). SPADE further learns to generate executable environments, using the agent's return difference with and without privileged hints as feedback for environment design (Liu et al., 2026a).

These developments build on self-training with successful generations in STaR and ReST-EM (Zelikman et al., 2022; Singh et al., 2023), demonstration-based self-play in SPIN (Chen et al., 2024b), and rationale learning in Quiet-STaR (Zelikman et al., 2024). SCoRe learns self-correction through multi-turn online RL (Kumar et al., 2025). Feedback itself can also improve: V-STaR trains inference-time verifiers (Hosseini et al., 2024), ReST-MCTS* jointly improves reasoning policies and process reward models through tree search (Zhang et al., 2024), and Self-Rewarding Language Models refine reward judgments through iterative preference optimization (Yuan et al., 2024). More recently, Sol-Ver co-trains code and test generation (Lin et al., 2026), while Self-Guide jointly trains an agent's policy and self-guidance that supplies internal rewards (Wang et al., 2026).

Several recent approaches assess generated practice using learning-potential or learner-progress signals. Eva uses a creator–solver bilevel hierarchy with a fixed reward oracle (Ye et al., 2025); SOAR rewards generated training data through measured short-run student improvement (Sundaram et al., 2026); and INFUSER uses optimizer-aware influence rewards to estimate local target-task benefit (Chen et al., 2026). Across these lines of work, curricula, verification mechanisms, and reward judgments can evolve during self-improvement. Our formulation specifically learns a shared incentive model governing both questioner and solver, evaluated through the equilibrium solver's target-task performance.

Bilevel optimization. Bilevel optimization evaluates an UL decision through the solution of an LL optimization problem, with applications to hyperparameter optimization and meta-learning (Franceschi et al., 2018). Classical approximation and hypergradient methods study strongly convex LL problems (Ghadimi & Wang, 2018; Ji et al., 2021), while implicit differentiation with approximate inverse-Hessian-vector products supports large-scale hyperparameter learning (Lorraine et al., 2020).

Value-gap penalty methods are particularly relevant to our approach because they encode LL optimality as a constraint violation. F^{2}SA (Kwon et al., 2023) uses a penalty surrogate and tracks the original and perturbed LL minimizers with first-order stochastic gradients under LL strong convexity. Kwon et al. (2024) extend the penalty analysis beyond strong convexity under uniform proximal error-bound and regularity conditions. Chen et al. (2025) develop two-time-scale updates for this first-order penalty approach, achieving near-optimal $\widetilde{O}(\epsilon^{-2})$ deterministic oracle complexity for gradient-norm ϵ-stationarity in the nonconvex–strongly-convex setting. Chen et al. (2024a) further analyze penalty-based first-order methods for nonconvex LL problems under uniform Polyak–Łojasiewicz conditions on the perturbed LL objectives. PBGD studies penalty reformulations, solution correspondence, and finite-time convergence under LL error bounds weaker than strong convexity (Shen & Chen, 2023). PBRL brings this perspective to bilevel RL and RLHF through value and Bellman penalties, together with differentiability and policy-gradient analysis (Shen et al., 2025). Collectively, these methods show how LL optimality gaps can supply first-order update directions while controlling finite-penalty approximation. We represent the LL game's two unilateral optimality conditions with playerwise best-response gaps, retaining them separately in policy updates and combining them in the incentive update.

Bilevel optimization with a lower-level game. When the LL consists of interacting players, the UL is constrained by their equilibrium response. Mathematical programs with equilibrium constraints provide a general framework (Luo et al., 1996). Li et al. (2023) develop hierarchy-free

approximations for bilevel programs with LL variational-inequality constraints through Cournot and monopoly formulations. BIG Hype tracks the LL equilibrium and its Jacobian to compute distributed hypergradient updates (Grontas et al., 2024). Samadi & Yousefian (2025) study regularization and inexact-projection methods for bilevel variational inequalities and optimal NE selection. For multiple-follower Stackelberg problems, Niu et al. (2025) use a Lagrangian reformulation to obtain fully first-order stationarity guarantees under strongly monotone follower games.

Incentive design is a natural application of this hierarchy: a designer changes agents' rewards to improve an external objective. Liu et al. (2022) jointly update incentives and agent strategies in a single loop, with global convergence guarantees under strong equilibrium stability, strong convexity of the composite UL objective, and additional regularity conditions. In RL, Yang et al. (2022) optimize incentives by differentiating through multi-agent learning updates. These learning-based approaches connect incentive selection to agents' adaptation without requiring that every practical update solve an exact LL equilibrium.

Penalty methods also apply directly to game-constrained bilevel RL. PBRL extends its framework to two-player zero-sum LL Markov games using a Nikaido–Isoda gap penalty (Shen et al., 2025). PANDA develops penalty-augmented Nikaido–Isoda descent–ascent for regularized zero-sum Markov games, using first-order policy gradients without second-order information (Zheng et al., 2026). Our formulation instead learns shared incentives above a general-sum questioner–solver game and evaluates them through the equilibrium solver's target performance. BiGAP uses both players' best-response gaps in the incentive update and player-specific penalties in policy updates. These policy-specific directions form a mixed game operator, generally distinct from the gradient of one aggregate penalty objective.

B Detailed Implementation of BiGAP

This appendix presents the practical implementation of BiGAP in execution order: question rollouts, solution rollouts, reward computation, policy training, incentive updating, and UL dataset evolution. The questioner π_ϕ and solver π_ψ have separate policy parameters, while x parameterizes their shared incentive model. All three parameter blocks are unconstrained. The UL dataset $\mathcal{D}_{\mathrm{UL}}$ supplies target questions for evaluating solver performance and seeds for related practice.

Each iteration collects observations before updating any model. The policy and incentive stages reuse these stored observations, and the dataset is held fixed throughout collection and training. We give the definitions, objectives, and estimators for each stage below. Algorithm 1 in Section 3 gives the complete procedure. Task-specific interfaces are given in Appendix B.7, and experimental settings are recorded in Appendix C.

B.1 Question rollouts

Prompt construction. At iteration t, we load the current parameters (ϕ_t, ψ_t, x_t). Let $\mathcal{U}$ denote the set of question-generation prompts and ν_{seed} their sampling distribution. The prompt batch $\mathcal{U}_t$ mixes two modes. Seed-conditioned prompts draw questions from $\mathcal{D}_{\mathrm{UL}}$ and request related, easier practice; exploration prompts request new questions without a particular seed. The prompt budget is B_{prompt}, with a configurable exploration-to-similarity ratio. All seeds are used when the resulting mixture fits the budget; otherwise, seeds are sampled without replacement and the two modes are allocated within the budget. Each exploration prompt has its own synthetic seed identifier, so it defines a separate advantage group even when its instruction text is shared with other exploration prompts. If $\mathcal{D}_{\mathrm{UL}}$ is empty, the entire budget is assigned to exploration.

Sampling and admissibility. For each prompt $s \in \mathcal{U}_t$, the questioner samples N_Q completions:

$$\mathcal{Q}_t(s) = (q_{t,s,j})_{j=1}^{N_Q}, \qquad q_{t,s,j} \sim \pi_{\phi_t}(\cdot \mid s). \tag{11}$$

All rollout collections are indexed by sample identifiers. Thus, repeated text and parsing failures remain separate question actions in their original groups; removing them would change the sampled advantage baseline. Let $\mathcal{Q}_t$ collect these actions, and let $s(q)$ record each action's originating prompt. The interface provides a format check $F_Q(q) \in \{0, 1\}$ and a question-validity check $V_Q(q) \in \{0, 1\}$.

The questions passed to the solver are

$$\mathcal{Q}_t^{\text{elig}} = \{q \in \mathcal{Q}_t : F_Q(q)V_Q(q) = 1\}. \tag{12}$$

Any reference answer or other information required to evaluate solutions is prepared and fixed before the training responses are sampled. An ineligible question receives a zero-reward questioner record and requires no solver rollout. The concrete parsing and reference-preparation procedures are given in Appendix B.7.

B.2 Solution rollouts

Lower- and upper-level response streams. The solver samples $N_S > 2$ responses per eligible generated question for LL practice and, independently, N_{UL} responses per UL question:

$$\begin{aligned} \mathcal{A}(q) &= (a_{t,q,j})_{j=1}^{N_S}, & a_{t,q,j} &\sim \pi_{\psi_t}(\cdot \mid q), & q &\in \mathcal{Q}_t^{\text{elig}}, \\ \mathcal{A}^{\text{UL}}(q) &= (a_{t,q,j}^{\text{UL}})_{j=1}^{N_{\text{UL}}}, & a_{t,q,j}^{\text{UL}} &\sim \pi_{\psi_t}(\cdot \mid q), & q &\in \mathcal{D}_{\text{UL}}. \end{aligned} \tag{13}$$

The iteration index t is implicit in the response collections $\mathcal{A}(q)$ and $\mathcal{A}^{\text{UL}}(q)$. Both streams use the same pre-update checkpoint π_{ψ_t}. Even when a question appears in both streams, its response batches are sampled independently. No responses are regenerated between the policy and incentive updates. An empty UL dataset contributes no UL response stream.

Verification and pass-rate estimation. The fixed task interface returns binary acceptance $v(q, a) \in \{0, 1\}$. Invalid or rejected responses have value zero. The empirical pass rates are

$$\widehat{\rho}_t(q) = \frac{1}{N_S} \sum_{j=1}^{N_S} v(q, a_{t,q,j}), \tag{14a}$$

$$\widehat{\rho}_t^{\text{UL}}(q) = \frac{1}{N_{\text{UL}}} \sum_{j=1}^{N_{\text{UL}}} v(q, a_{t,q,j}^{\text{UL}}). \tag{14b}$$

Every attempted response remains in its stream's denominator, including malformed outputs. These outcomes are saved before any optional training filter is applied, so training eligibility does not change the recorded pass rates or evaluation losses.

Upper-level objective and estimate. For a fixed nonempty $\mathcal{D}_{\text{UL}}$, the task-failure objective is

$$f(x, \phi, \psi) = \mathbb{E}_{q \sim \mathcal{D}_{\text{UL}},\, a \sim \pi_\psi(\cdot|q)}[1 - v(q, a)], \tag{15}$$

where questions are sampled uniformly. This objective evaluates the incentives through the solver they induce, but has no direct dependence on x or ϕ when the solver and question distribution are fixed. Its estimate at the collection checkpoint is

$$\widehat{f}_t(x_t, \phi_t, \psi_t) = \frac{1}{|\mathcal{D}_{\text{UL}}|} \sum_{q \in \mathcal{D}_{\text{UL}}} [1 - \widehat{\rho}_t^{\text{UL}}(q)]. \tag{16}$$

We abbreviate this recorded scalar by $\widehat{f}_t$. Each attempted UL response incurs loss $1 - v(q, a)$ independently of the learned reward. No UL loss is estimated when $\mathcal{D}_{\text{UL}}$ is empty.

B.3 Reward computation

Gates and trainable neural scores. Using the question checks above and a response-format check $F_S(a)$, define

$$G_Q(s, q, a) = F_Q(q)V_Q(q), \qquad G_S(s, q, a) = F_Q(q)V_Q(q)F_S(a)v(q, a). \tag{17}$$

The questioner gate requires an admissible question; the solver gate also requires an accepted response. In particular, an invalid question gives zero solver rewards for its entire response group, hence no reward-based group-relative signal. The implementation avoids generating these response groups through equation 12.

For role $i \in \{Q, S\}$, the neural incentive component is

$$h_{x,i}(s, q, a) = \text{sigmoid}(\ell_x(\text{prompt}_i(s, q, a))) \,. \tag{18}$$

Here, ℓ_x is a shared sequence classifier with one scalar output, and $\text{prompt}_Q, \text{prompt}_S$ are role-specific templates that may include the seed, question, solution, and reference answer. The composed reward is

$$r_{x,i}(s, q, a) = G_i(s, q, a) P_i(L_i) \left[c_{i,\text{cls}} h_{x,i}(s, q, a) + \sum_{k \in \mathcal{K}_i} c_{i,k} r_{i,k}(s, q, a) \right]. \tag{19}$$

The set $\mathcal{K}_Q$ contains the selected frozen question-quality components. Our main neural configuration uses $\mathcal{K}_Q = \{\text{sim}, \text{div}, \text{diff}\}$ and $\mathcal{K}_S = \varnothing$: the solver reward is simply $G_S P_S c_{S,\text{cls}} h_{x,S}$, with no auxiliary reward component. The coefficients $c_{i,\text{cls}}, c_{i,k}$ are fixed configuration values and receive no gradient. Classifier training updates its trainable backbone parameters or LoRA parameters together with its scalar head. The weighted sum of reward components need not lie in $[0, 1]$, even though each classifier probability and each built-in quality component does.

Fixed length-penalty multiplier. Let L_Q measure question length and L_S measure solution length. For a role's output length L, the fixed multiplier is

$$P_i(L) = \begin{cases} 1, & L \le L_i^{\text{tgt}}, \\ \left(\dfrac{L - L_i^{\text{tgt}}}{L_i^{\text{ref}} - L_i^{\text{tgt}}} \right)^{p_{\text{len}}} & \\ b_{\text{len}} & , \quad L > L_i^{\text{tgt}}. \end{cases} \tag{20}$$

Here, $L_i^{\text{ref}} > L_i^{\text{tgt}}$, $b_{\text{len}} \in [0, 1]$, and $p_{\text{len}} > 0$. Thus, $P_i = 1$ through the target length and equals b_{len} at the reference length; decay continues beyond that reference, which is not a hard generation cutoff. The special setting $b_{\text{len}} = 0$ gives a hard over-target cutoff, and disabling the multiplier sets $P_i = 1$. Length can be measured on the complete response, including reasoning, or on its parsed task content, using characters, whitespace-delimited tokens, or model tokens. These length units and thresholds are independent of the generation and training token budgets. The multiplier P_i remains fixed during incentive training.

Frozen quality components. Difficulty rewards questions near a target pass rate $\tau_{\text{diff}} \in (0, 1)$. Using the LL estimate $\widehat{\rho}_t(q)$ in equation 14, define

$$r_{Q,\text{diff}}(q) = \begin{cases} 1 - \left((\widehat{\rho}_t(q) - \tau_{\text{diff}}) / \tau_{\text{diff}} \right)^2, & \widehat{\rho}_t(q) \le \tau_{\text{diff}}, \\ 1 - \left((\widehat{\rho}_t(q) - \tau_{\text{diff}}) / (1 - \tau_{\text{diff}}) \right)^2, & \widehat{\rho}_t(q) > \tau_{\text{diff}}. \end{cases} \tag{21}$$

It is zero for uniformly failed or uniformly successful response groups and maximal at τ_{diff}. It applies to both question-generation modes. For exploration questions, diversity combines novelty relative to a reference corpus with rarity among eligible exploration questions generated in the current iteration:

$$r_{Q,\text{div}}(q) = \alpha_{\text{div}} \, \text{clip}_{[0,1]} \left(1 - \max_j \langle e(q), \mu_j \rangle \right) + (1 - \alpha_{\text{div}}) \frac{1}{n_{c(q)}}. \tag{22}$$

We use normalized question embeddings $e(q)$ from a frozen model and deterministic k-means. The reference corpus is the current UL dataset $\mathcal{D}_{\text{UL}}$, held fixed within each iteration; clustering its questions yields normalized centers μ_j for assessing novelty. Separately clustering all exploration questions in $\mathcal{Q}_t^{\text{elig}}$ across the iteration's prompts gives assignments $c(q)$ and cluster sizes $n_{c(q)}$. Each question identifier is counted once; assignments and counts remain fixed across training minibatches. The weight $\alpha_{\text{div}} \in [0, 1]$ balances novelty and rarity. When $\mathcal{D}_{\text{UL}}$ is empty, novelty is set to one; diversity is zero for non-exploration questions.

Rarity is a heuristic: its mean over N clustered questions is K_{occ}/N, where K_{occ} counts occupied clusters, and is constant when both counts are fixed.

For seed-conditioned questions, a frozen similarity judge scores a meaningful extension of the seed while discouraging exact repetition:

$$r_{Q,\text{sim}}(s, q) = \mathbf{1}\{s \text{ is seed-conditioned}\} \; \text{Judge}_{\text{sim}}(s, q), \qquad \text{Judge}_{\text{sim}}(s, q) \in [0, 1]. \tag{23}$$

This is a rubric-based model score, rather than embedding similarity. Optional configurations also provide a frozen clarity judge for question formulation and a frozen solver-quality judge for the question and full response. The latter is an alternative reward component, not part of the main solver reward with $\mathcal{K}_S = \varnothing$. These judges provide soft quality signals; task verification supplies the correctness gates. Their saved outputs are reused during incentive training. The pass rates and cluster occupancies are computed from the iteration's collected rollouts and held fixed during questioner optimization. The resulting policy surrogate assigns each question its recorded reward; it does not differentiate through other questions' sampling probabilities or through recomputation of the batch statistics.

Rewards assigned to policy actions. The questioner receives the mean interaction reward over a question's sampled responses, while the solver receives a reward for each response:

$$R_{x,Q}(s,q) = \begin{cases} |\mathcal{A}(q)|^{-1} \displaystyle\sum_{a \in \mathcal{A}(q)} r_{x,Q}(s,q,a), & q \in \mathcal{Q}_t^{\text{elig}}, \\ 0, & q \notin \mathcal{Q}_t^{\text{elig}}, \end{cases} \qquad R_{x,S}(q,a) = r_{x,S}(s(q),q,a). \tag{24}$$

Policy training uses these rewards evaluated at the collection parameters x_t. Incentive training later recomputes their learned components on the same observations under the candidate parameters x. Optional task-specific gates can further restrict the training records, as detailed in Appendix B.7.

B.4 Policy training

Population objectives and policy targets. Under the interaction distribution $s \sim \nu_{\text{seed}}$, $q \sim \pi_\phi(\cdot \mid s)$, and $a \sim \pi_\psi(\cdot \mid q)$, the regularized LL values are

$$J_Q(x,\phi,\psi) = \mathbb{E}_{s,q,a}\left[r_{x,Q}(s,q,a) - \tau_Q \log \frac{\pi_\phi(q \mid s)}{\pi_Q^{\text{ref}}(q \mid s)} \right],$$
$$J_S(x,\phi,\psi) = \mathbb{E}_{s,q,a}\left[r_{x,S}(s,q,a) - \tau_S \log \frac{\pi_\psi(a \mid q)}{\pi_S^{\text{ref}}(a \mid q)} \right]. \tag{25}$$

Here, $\pi_Q^{\text{ref}}, \pi_S^{\text{ref}}$ are fixed reference policies, $\tau_Q, \tau_S > 0$, and the loss convention is $g_i = -J_i$. The interaction expectation includes any additional responses used to compute rewards; fixed iteration-level rollout context is suppressed in the notation. For the idealized penalty construction, assume the best-response values $J_Q^\star(x,\psi) = \max_{\phi'} J_Q(x,\phi',\psi)$ and $J_S^\star(x,\phi) = \max_{\psi'} J_S(x,\phi,\psi')$ are attained. Since $g_i = -J_i$, these values satisfy $J_i^\star = -g_i^\star$ for $i \in \{Q,S\}$. Thus, the gaps $\delta_Q = J_Q^\star - J_Q$ and $\delta_S = J_S^\star - J_S$ are exactly the unilateral optimality gaps in equation 6. With $\lambda > 0$, the policy targets are

$$\mathcal{L}_{\lambda,Q}(x,\phi,\psi) = f(x,\phi,\psi) - \lambda J_Q(x,\phi,\psi) + \lambda J_Q^\star(x,\psi),$$
$$\mathcal{L}_{\lambda,S}(x,\phi,\psi) = f(x,\phi,\psi) - \lambda J_S(x,\phi,\psi) + \lambda J_S^\star(x,\phi). \tag{26}$$

The questioner minimizes $\mathcal{L}_{\lambda,Q}$ over ϕ, and the solver minimizes $\mathcal{L}_{\lambda,S}$ over ψ, holding the other parameter blocks fixed. Each best-response term is constant in its player's update. The objective f in equation 15 is also independent of ϕ for the fixed solver and UL dataset. Thus, questioner learning reduces to maximizing J_Q, while solver learning combines LL rewards with UL task feedback. Practical training approximates these targets with the policy losses below, without solving a best-response problem.

Returns and group-relative advantages. We use the same penalty coefficient $\lambda > 0$ for the questioner, solver, and incentive objectives. With the observed rewards in equation 24, the policy returns are

$$R_Q = \lambda R_{x_t,Q}(s,q), \qquad R_S^{\text{LL}} = \lambda R_{x_t,S}(q,a), \qquad R_S^{\text{UL}} = v(q,a) - 1. \tag{27}$$

The last return is negative task failure. All returns and verification outcomes remain fixed during policy optimization. For a sample j in group $\mathcal{G}$, the leave-one-out advantage is

$$A_j = \begin{cases} R_j - \dfrac{1}{|\mathcal{G}|-1} \sum_{k \in \mathcal{G} \setminus \{j\}} R_k, & |\mathcal{G}| > 1, \\ R_j, & |\mathcal{G}| = 1. \end{cases} \tag{28}$$

Questioner groups share a seed identifier; solver groups share a question identifier, separately within the lower- and upper-level streams. No standard-deviation normalization is applied, so this linear baseline preserves the common reward multiplier λ.

Policy loss and reference regularization. Both policies use a variant of group relative policy optimization (GRPO). Let y_j be the token sequence of sampled completion j, and let $H_{j,k}$ contain its prompt and preceding completion tokens. For a role $i \in \{Q, S\}$, let $\vartheta = \phi$ when $i = Q$ and $\vartheta = \psi$ when $i = S$. With KL coefficient τ_i^{opt}, we write the per-response policy surrogate using advantage-weighted token log-probabilities:

$$\ell_i(j;\vartheta) = \frac{1}{|y_j|}\sum_{k=1}^{|y_j|}\left[-A_j \log \pi_\vartheta(y_{j,k} \mid H_{j,k}) + \tau_i^{\mathrm{opt}}\kappa_{j,k}\right],$$
$$\kappa_{j,k} = \exp(d_{j,k}) - d_{j,k} - 1, \qquad d_{j,k} = \log \pi_i^{\mathrm{ref}}(y_{j,k} \mid H_{j,k}) - \log \pi_\vartheta(y_{j,k} \mid H_{j,k}). \tag{29}$$

The fixed reference π_i^{ref} is the role's configured original base policy. Reference token probabilities and advantages are fixed; current token log-probabilities and the KL term are recomputed at every optimization step. Prompt and padding tokens are masked, and completion tokens are averaged within each response before responses are averaged. Tokenization retains the full prompt and the completion prefix fitting the training budget; likelihoods use exactly those retained tokens. The implementation coefficient τ_i^{opt} is configured independently of λ. Matching the penalty-scaled KL convention for J_i in equation 25 requires $\tau_i^{\mathrm{opt}} = \lambda\tau_i$ for a single lower-level stream; the implementation does not automatically multiply this coefficient by λ.

Combining policy losses. Let $\mathcal{B}_Q$ contain all sampled question actions, and let $\mathcal{B}_S^{\mathrm{LL}}, \mathcal{B}_S^{\mathrm{UL}}$ contain the solver-training records after any task-specific training gates. The empirical policy losses are

$$\widehat{\mathcal{L}}_{\lambda,Q}^{\mathrm{policy}}(\phi) = \frac{1}{|\mathcal{B}_Q|}\sum_{j\in\mathcal{B}_Q}\ell_Q(j;\phi),$$
$$\widehat{\mathcal{L}}_{\lambda,S}^{\mathrm{policy}}(\psi) = \frac{1}{|\mathcal{B}_S^{\mathrm{LL}}|}\sum_{j\in\mathcal{B}_S^{\mathrm{LL}}}\ell_S^{\mathrm{LL}}(j;\psi) + \frac{1}{|\mathcal{B}_S^{\mathrm{UL}}|}\sum_{j\in\mathcal{B}_S^{\mathrm{UL}}}\ell_S^{\mathrm{UL}}(j;\psi). \tag{30}$$

The two solver losses use equation 29 with their respective stream's advantages. Each includes its configured KL term. Finite optimization steps on the nonempty training objectives produce the updated policy checkpoints. Empty solver streams contribute no term; if neither contains training examples, $\psi_{t+1} = \psi_t$.

Separately averaging the solver streams keeps their relative weight independent of their sample counts. When both streams are present, the smaller stream is shuffled and cycled to match the larger, with sample weights implementing this sum of means. For computational efficiency, solver rollouts are collected only for eligible generated questions, and the LL loss is normalized over their responses. This conditioning changes the effective LL weighting relative to the population objective and omits reference regularization on ineligible questions. The UL KL term is additional practical regularization beyond $f - \lambda J_S$. These choices, together with independently configured KL coefficients and optional training filters, make equation 30 a practical surrogate for the population targets in equation 26.

B.5 Incentive update

Incentive objective. Whereas each policy follows its own gap penalty, the idealized incentive block accounts for both players by minimizing

$$\mathcal{L}_\lambda(x,\phi,\psi) = f(x,\phi,\psi) + \lambda\big[\delta_Q(x,\phi,\psi) + \delta_S(x,\phi,\psi)\big] \tag{31}$$

over x, with both policies held fixed. Under envelope regularity, the incentive derivative of a gap compares reward derivatives at a best response and under current play, without differentiating through the response policy. The reference regularizers have no explicit dependence on x. We approximate these comparisons using stored candidate rewards.

Finite candidate gaps. For the questioner, a context is a generation prompt s with candidates $\mathcal{Q}_t(s)$; for the solver, it is an eligible question q with candidates $\mathcal{A}(q)$. In the generic notation of Section 3, these collections are $\mathcal{C}_Q = \mathcal{U}_t$, $\mathcal{Y}_Q(s) = \mathcal{Q}_t(s)$, $\mathcal{C}_S = \mathcal{Q}_t^{\text{elig}}$, and $\mathcal{Y}_S(q) = \mathcal{A}(q)$, with the iteration index suppressed. Using the action rewards in equation 24, define

$$\widehat{\delta}_Q(x) = \frac{1}{|\mathcal{U}_t|} \sum_{s \in \mathcal{U}_t} \left[\max_{q \in \mathcal{Q}_t(s)} R_{x,Q}(s,q) - \frac{1}{|\mathcal{Q}_t(s)|} \sum_{q \in \mathcal{Q}_t(s)} R_{x,Q}(s,q) \right],$$

$$\widehat{\delta}_S(x) = \frac{1}{|\mathcal{Q}_t^{\text{elig}}|} \sum_{q \in \mathcal{Q}_t^{\text{elig}}} \left[\max_{a \in \mathcal{A}(q)} R_{x,S}(q,a) - \frac{1}{|\mathcal{A}(q)|} \sum_{a \in \mathcal{A}(q)} R_{x,S}(q,a) \right]. \tag{32}$$

Here $\mathcal{U}_t$ indexes the sampled prompt batch by question-generation prompt identifiers, keeping repeated prompt text as separate instances. Each role is averaged uniformly over its own contexts, and the two role means are summed; an absent role contributes zero. Equivalently, $\widehat{\delta}_i(x) = \widehat{R}_i^{\text{best}}(x) - \widehat{R}_i(x)$, where $\widehat{R}_i^{\text{best}}$ averages the best reward in each context and $\widehat{R}_i$ averages the within-context mean rewards. Failed response gates remain zero-valued candidates; contexts with one candidate have zero gap. These are differences between empirical composed rewards, not estimates of the full KL-regularized values J_Q, J_S. In particular, finite candidate maximization neither optimizes a response policy nor includes a KL cost for changing its action distribution.

Optimization on fixed observations. The practical incentive loss is

$$\widehat{\mathcal{L}}_\lambda(x) = \widehat{f}_t + \lambda \left[\widehat{\delta}_Q(x) + \widehat{\delta}_S(x) \right], \tag{33}$$

where $\widehat{f}_t$ is the recorded mean upper-level verification failure. The data, frozen components, correctness gates, and length-penalty factors remain fixed throughout this update. Each trajectory's recomputed reward has the form $\text{offset} + \text{scale} \cdot h_{x,i}$, with only the classifier term differentiated. At every optimization step, maximizers are recomputed under the current x, their discrete indices are detached, and gradients flow through both selected rewards and context means. Classifier batches keep all candidates of a context together and use role-balancing weights to preserve the sum of role-wise means.

Since $\widehat{f}_t$ is constant with respect to x, it supplies no direct gradient in this stage. Nor does the implementation differentiate through the policy updates, verification, or dataset selection. The update therefore uses the finite gap penalty as a practical incentive surrogate. An empty upper-level bootstrap has no measured $\widehat{f}_t$; in that case the incentive checkpoint is explicitly carried forward unchanged.

B.6 UL DATASET EVOLUTION

Initialization and timing. The UL dataset can start empty or with unlabeled questions; any information required by the task's acceptance rule must be prepared before UL response collection. When initialized empty, the framework uses exploration-only question generation and LL-only policy training. Incentive parameters are carried forward unchanged until a nonempty UL dataset is available.

After the model updates, the implementation can refresh $\mathcal{D}_{\text{UL}}$ using the stored pass rates in equation 14. All admission and removal decisions use the pre-update solver's outcomes; the updated solver is not evaluated again for this stage. The refreshed questions provide UL feedback and generation seeds in the next iteration. This curriculum is an implementation choice outside the fixed-distribution formulation and analysis.

Admission and retention. Generated questions are eligible for admission when

$$v_{\text{add}}^- < \widehat{\rho}_t(q) \le v_{\text{add}}^+, \tag{34}$$

in addition to passing question-format and task-eligibility checks. The strict lower bound and inclusive upper bound are part of the filter. Candidates are ordered by increasing pass rate, and at most M_{add} are added. Unicode- and whitespace-normalized text is used to remove duplicates of existing seeds

and previously selected candidates. This deduplication affects dataset admission rather than the rollout action groups. The selected candidates form $\mathcal{D}_{\mathrm{UL}}^{\mathrm{add}}$. The deletion interval typically retires easy questions. With both endpoints included, the retained set is

$$\mathcal{D}_{\mathrm{UL}}^{\mathrm{keep}} = \{q \in \mathcal{D}_{\mathrm{UL}} : \widehat{\rho}_t^{\mathrm{UL}}(q) \notin [v_{\mathrm{del}}^-, v_{\mathrm{del}}^+]\}. \tag{35}$$

The optional accumulation setting instead takes $\mathcal{D}_{\mathrm{UL}}^{\mathrm{keep}} = \mathcal{D}_{\mathrm{UL}}$. When dataset updates are enabled, the refresh is

$$\mathcal{D}_{\mathrm{UL}} \leftarrow \mathcal{D}_{\mathrm{UL}}^{\mathrm{keep}} \cup \mathcal{D}_{\mathrm{UL}}^{\mathrm{add}}. \tag{36}$$

Otherwise, the dataset remains unchanged. In every case, it is held fixed again during the next iteration's rollout collection and model updates.

Execution and saved state. The pipeline persists question and response identifiers, behavior-policy versions, raw and parsed outputs, verification outcomes, reward components, and iteration checkpoints. Generation and reward inference use configurable inference backends; policy and neural incentive updates use the corresponding training backends, with LoRA or full-parameter training. Stages execute in the order shown in Algorithm 1. Although the three training stages are scheduled sequentially, they reuse the same pre-update observations. Independent benchmark evaluation follows the updates and is not used as an incentive-training target. The input dataset and resulting version make each dataset update reproducible from its saved outcomes.

B.7 Task-specific implementation

The preceding stages use a common interface for parsing questions, preparing evaluation information, checking validity, and verifying responses. We instantiate that interface for mathematical question answering and formal theorem proving below.

Mathematical question answering: reference preparation. A generated question must have a usable reference answer before the ordinary solver responses in equation 13 are collected. In reference mode, the parser uses the answer supplied with the question. In self-consensus mode, the interface first samples a separate batch of M_{qa} responses from the frozen current solver and groups equivalent parsed answers into classes $\mathcal{C}_k(q)$. It selects the largest class and accepts its answer as a reference only when

$$k^* \in \arg\max_k |\mathcal{C}_k(q)|, \qquad \frac{|\mathcal{C}_{k^*}(q)|}{M_{\mathrm{qa}}} \geq h_{\mathrm{qa}} > \frac{1}{2}. \tag{37}$$

The denominator includes all labeling attempts, including malformed ones. The accepted reference is frozen before the independent training responses are generated. These references are supplied labels or consensus estimates, rather than certificates of correctness. Questions entering the UL stream carry supplied or previously frozen reference answers; the current QA UL interface requires this reference metadata.

Mathematical question answering: verification and training gates. The interface extracts each response's answer and assigns $v(q, a) = 1$ when it matches the fixed reference under the configured exact or mathematical matcher. Failed parsing or disagreement gives $v(q, a) = 0$. An optional reference-match training gate requires at least one ordinary solver response to match the reference. For unsupported generated questions, the questioner reward is set to zero and the entire response group is excluded from solver training. The same group-level gate can exclude unsupported UL questions from solver training. Unlike pre-response question eligibility, this gate uses sampled outcomes and changes the training distributions. It leaves the stored verification outcomes, pass-rate denominators, and UL loss estimate unchanged. Absence of a sampled match does not establish that a reference answer is false. Held-out accuracy uses benchmark references and the declared answer-selection rule.

Formal theorem proving. The question parser requires a statement-only Lean theorem in the prescribed format. Question verification checks whether the statement elaborates in the configured Lean/Mathlib environment; this result is saved and reused for reward gates and dataset admission. The solver supplies a candidate proof, and proof checking determines $v(q, a)$. A failed or malformed

proof has value zero. Successful statement elaboration establishes that the question is well formed, but does not establish provability. Held-out pass@k measures success within the declared attempt budget and proof-checking timeout.

C Experimental Details

Table 4 lists the BiGAP implementation settings.

Table 4: BiGAP algorithm parameters for mathematical QA and formal theorem proving. $Q/S/x$ denote the questioner, solver, and incentive-model blocks; T is the configured horizon, not the number of reported checkpoints.

Parameter	Meaning	Math QA	Theorem proving
Core algorithm and sampling			
T	Configured outer-iteration horizon	50	50
$\mathcal{D}_{\mathrm{UL}}^{0}$	Initial UL dataset	Empty	Seeded (unlabeled)
B_{prompt}	Maximum prompt groups per iteration	1,000	1,000
N_Q	Questions sampled per prompt group	8	8
N_S	LL solutions per generated question	8	8
N_{UL}	UL solutions per question	16	16
Prompt mixture	Exploration : similarity	1 : 1	1 : 1
θ_Q, θ_S	Sampling temperatures	1.0/1.0	1.0/1.0
p_Q, p_S	Nucleus-sampling probabilities	0.95/0.90	0.95/0.90
$L_Q^{\mathrm{gen}}, L_S^{\mathrm{gen}}$	Maximum generated tokens	8,192/4,096	4,096/4,096
Block scales ($Q/S/x$)	Configured update multipliers	0.1/0.1/1.0	0.1/0.1/1.0
$\tau_Q^{\mathrm{opt}}, \tau_S^{\mathrm{opt}}$	Policy-objective KL coefficients	0.01/0.01	0.01/0.01
Reward construction			
$c_{Q,\mathrm{cls}}$	Questioner learned-score coefficient	1.0	1.0
$c_{Q,\mathrm{diff}}, c_{Q,\mathrm{div}}, c_{Q,\mathrm{sim}}$	Difficulty/diversity/similarity coefficients	0.5/0.5/0.5	0.5/0.5/0.5
$c_{S,\mathrm{cls}}$	Solver learned-score coefficient	1.0	1.0
τ_{diff}	Target empirical solver pass rate	0.5	0.5
α_{div}	Novelty weight in diversity reward	0.7	0.7
K_{div}	Maximum diversity clusters	8	8
Clustering steps	Number of k-means iterations	25	25
$L_i^{\mathrm{tgt}}, L_i^{\mathrm{ref}}$	Target/reference lengths, $i \in \{Q, S\}$	3,072/4,096	3,072/4,096
$b_{\mathrm{len}}, p_{\mathrm{len}}$	Length-decay base/exponent	0.5/1.0	0.5/1.0
Curriculum and task verification			
$(v_{\mathrm{add}}^{-}, v_{\mathrm{add}}^{+}]$	Admission interval for $\widehat{\rho}_t(q)$	$(0, 0.3]$	$(0, 0.3]$
M_{add}	Maximum additions per iteration	100	200
$[v_{\mathrm{del}}^{-}, v_{\mathrm{del}}^{+}]$	Removal interval for $\widehat{\rho}_t^{\mathrm{UL}}(q)$	$[0.8, 1.0]$	$[0.8, 1.0]$
Task verifier	Training-time acceptance checker	`math_verify`	Lean 4

The prompt-mixture row specifies the exploration-to-similarity ratio within ν_{seed}, the distribution over generation prompts in Section 3. The supplied configurations record separate questioner/solver/incentive-model update scales of $0.1/0.1/1.0$. We report these as block scales rather than identifying them with the common penalty parameter λ in equation 27 and equation 33. The fixed coefficients $c_{i,\mathrm{cls}}, c_{i,k}$ weight reward components; only the incentive-model parameters x are learned. The policy-training KL coefficients τ_i^{opt} follow equation 29.

Benchmark provenance. MATH-500 is the 500-problem test split released with PRM800K (Lightman et al., 2023), drawn from MATH (Hendrycks et al., 2021). The AIME24/25/26 citations identify public releases of the corresponding Mathematical Association of America contest problems (Maxwell-Jia, 2024; MathArena, 2025; 2026); the latter two releases are maintained by MathArena (Dekoninck et al., 2026). The theorem-proving benchmark sources are miniF2F (Zheng et al., 2022), ProofNet (Azerbayev et al., 2023), ProverBench introduced with DeepSeek-Prover-V2 (Ren et al., 2025), MathOlympiadBench introduced with Goedel-Prover-V2 (Lin et al., 2025), and CAM-Bench (Long et al., 2026).

C.1 Evaluation Protocol

For both mathematical QA and formal theorem proving, we sample 16 responses per question on every benchmark, using temperature 0.7, nucleus-sampling probability 0.95, and thinking mode enabled. The configured maximum completion length is 8192 tokens, excluding the prompt. The QA evaluation additionally uses a 4096 token completion limit, yielding the two budgets reported in Table 2. All results are averaged over 3 random runs.

Mathematical QA. We extract the last complete `\boxed{...}` answer after the reasoning segment and compare it with the benchmark's supplied reference answer. The comparison normalizes spaces, math delimiters, and trailing periods, and recognizes equivalent integer, decimal, and fractional forms. A response without an extractable boxed answer is counted as incorrect. We report average response accuracy: for a benchmark with m questions and c_i correct responses among the 16 samples for question i, the percentage accuracy is

$$\text{Accuracy} = \frac{100}{m} \sum_{i=1}^{m} \frac{c_i}{16}.$$

Formal theorem proving. We extract the generated Lean proof body and combine it with the benchmark theorem statement. The proof-format check rejects generated bodies containing `sorry`, `admit`, `axiom`, or `unsafe`, as well as added top-level declarations or imports. A response is successful when it passes this check and the combined source compiles successfully under Lean 4. From the 16 responses per question, we report pass@4 and pass@16 using

$$\text{pass@}\, k = \frac{100}{m} \sum_{i=1}^{m} \left[1 - \frac{\binom{16-c_i}{k}}{\binom{16}{k}} \right], \qquad k \in \{4, 16\},$$

where c_i is the number of successful proofs for question i and $\binom{a}{k} = 0$ for $a < k$. Thus pass@16 is the percentage of questions with at least one successful proof among the 16 responses.

C.2 Additional Evaluation-Budget Analysis

Section 5.4 and Figure 1 present the iteration trends for QA at 8,192 tokens and proving at pass@16. At the additional QA limit of 4,096 tokens, BiGAP macro accuracy moves from 22.48% initially to 23.75%, 24.56%, 24.93%, 25.61%, and 24.53% at iterations 1–5, peaking at $t = 4$. For proving pass@4, the corresponding trajectory is 9.16% initially, followed by 10.17%, 10.09%, 10.46%, 10.20%, and 10.30%, with its peak at $t = 3$. BiGAP leads fixed rewards on both QA macro scores at every trained checkpoint and on both proving macro scores except at $t = 2$. At the last reported iteration ($t = 5$), it exceeds fixed rewards on seven of eight QA benchmark–budget pairs and six of ten proving pairs.

The best macro advantage persists at both QA completion limits (+1.31/+0.98 pp) and is larger at proving pass@16 than pass@4 (+0.66 versus +0.20 pp). The final macro scores remain above initialization but below each setting's observed peak. Holding the BiGAP QA checkpoint at $t = 3$ fixed, increasing the completion limit from 4,096 to 8,192 tokens raises macro accuracy by 19.97 pp. This budget dependence helps interpret score differences; it does not establish an efficiency advantage, and the five-checkpoint trajectories do not establish convergence.

C.3 Similarity Generation and Solver Feedback

Scope and aggregation. These diagnostics use the first five iterations of a separate QA trajectory, without assumed checkpoint alignment or shared settings with Table 2. Of 24,000 generated questions, 22,865 pass the pre-response eligibility checks in equation 12 and have complete groups of $N_S = 8$ LL responses. Mixed outcomes mean 1–7 matches to an unverified generated reference. Question-level reward $R_{x_t,Q}$ averages the eight interaction rewards; a category's reward share is its sum of question-level rewards divided by the total for that iteration.

Similarity generation supplies more mixed-outcome practice. Between iterations 2 and 5, similarity generation (conditioned on UL questions) increases its mixed-outcome fraction from

49.59% to 60.88%, versus 23.73% to 26.90% for exploration (Figure 2a). The source mixture stays 50/50 over iterations 2–5. Mixed groups provide nonuniform solver feedback and receive 57.39% to 68.06% of questioner reward, compared with 36.41% to 43.74% of analyzed questions. This supports the intended coupling between similarity generation and solver feedback; late rewards are dominated by difficulty and diversity terms, limiting attribution to learned incentives.

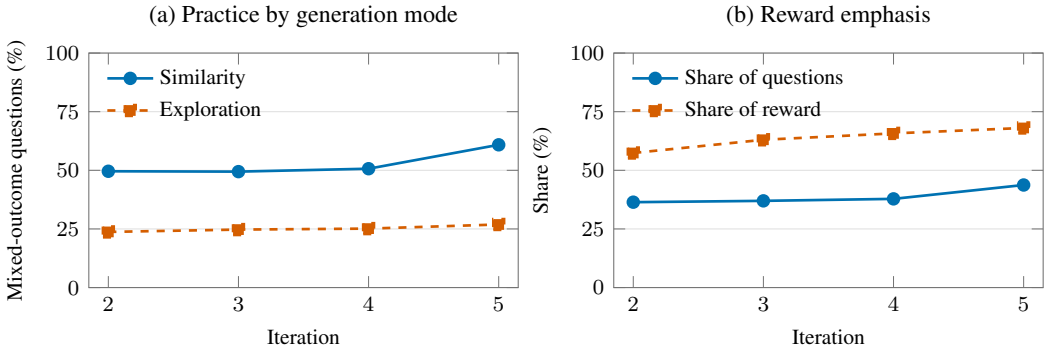


Figure 2: First-five-iteration QA diagnostics, restricted to complete $N_S = 8$ LL response groups. Mixed outcomes mean 1–7 reference matches. (a) Rates by generation mode. (b) Shares of analyzed questions and questioner reward $R_{x_t,Q}$, the mean interaction reward per question. Iterations 2–5 use equal source counts; iteration 1 is exploration-only. Self-generated references are unverified.

Table 5: Generation-source diagnostics. Analyzed questions pass eligibility checks and have eight LL responses; mixed fractions use the analyzed count. Iteration 1 is exploration-only. Iterations 2–5 generate equal numbers from each source.

	Generated	Exploration			Similarity		
Iteration	per source	Analyzed	Mixed	Mixed (%)	Analyzed	Mixed	Mixed (%)
2	800	767	182	23.73	738	366	49.59
3	1,600	1,541	381	24.72	1,515	749	49.44
4	2,400	2,299	578	25.14	2,267	1,149	50.68
5	3,200	3,089	831	26.90	3,034	1,847	60.88

Table 5 gives the counts underlying Figure 2a. Similarity generation consistently produces a larger mixed-outcome fraction than exploration under an equal source allocation. This characterizes the feedback distribution; the sources receive different inputs, so it does not isolate the effect of incentive learning.

D Theoretical Details and Proofs

Let $z = (x, \phi, \psi) \in Z = \mathbb{R}^{d_x} \times \mathbb{R}^{d_\phi} \times \mathbb{R}^{d_\psi}$. Unless otherwise specified, parameter-space norms are Euclidean. The original bilevel problem is

$$\begin{aligned} &\min_{(x,\phi,\psi)\in Z} f(x,\phi,\psi) \\ &\quad \text{s.t. } \phi \in \arg\min_{\phi'\in\mathbb{R}^{d_\phi}} g_Q(x,\phi',\psi), \\ &\quad\quad \psi \in \arg\min_{\psi'\in\mathbb{R}^{d_\psi}} g_S(x,\phi,\psi'), \end{aligned}$$

where

$$g_Q(x,\phi,\psi) = -\mathbb{E}_{s,q,a}\left[r_{x,Q}(s,q,a) - \tau_Q \log \frac{\pi_\phi(q|s)}{\pi_\phi^0(q|s)}\right],$$

$$g_S(x,\phi,\psi) = -\mathbb{E}_{s,q,a}\left[r_{x,S}(s,q,a) - \tau_S \log \frac{\pi_\psi(a|q)}{\pi_\psi^0(a|q)}\right].$$

The expectations use $s \sim \nu$, $q \sim \pi_\phi(\cdot \mid s)$, and $a \sim \pi_\psi(\cdot \mid q)$.

Algorithm 2 Exact-Oracle Anchored BiGAP

Require: $z_0 = (x_0, \phi_0, \psi_0) \in \mathbb{R}^{d_z}$, $w_0 = z_0$, $\lambda, \eta, \rho, \beta > 0$, horizon T.
1: **for** $t = 0, \dots, T-1$ **do**
2: Compute $\phi_t^{\mathrm{br}} \in \arg\min_{\phi'} g_Q(x_t, \phi', \psi_t)$ and $\psi_t^{\mathrm{br}} \in \arg\min_{\psi'} g_S(x_t, \phi_t, \psi')$.
3: Evaluate $\mathcal{H}_\lambda(z_t)$ in equation 7, using these best responses for the incentive block.
4: $z_{t+1} = z_t - \eta \mathcal{F}_\lambda(z_t, w_t)$; $w_{t+1} = w_t + \beta(z_{t+1} - w_t)$.
5: **end for**
6: **return** $z_{\widehat{t}}$, where $\widehat{t} \in \arg\min_{0 \le t < T} \|\mathcal{H}_\lambda(z_t)\|$.

Lemma D.1. *Let $u \in \mathbb{R}^n$ be a vector of utilities, and let $p^0 \in \Delta_n$ be a prior distribution with $p_j^0 > 0$ for every j. With the convention $0 \log(0/p_j^0) = 0$, for any $\tau > 0$, the unique global maximizer of the following optimization problem*

$$\max_{p \in \Delta_n} F(p) \triangleq \sum_{j=1}^n p_j u_j - \tau \sum_{j=1}^n p_j \log \frac{p_j}{p_j^0},$$

is given by the Gibbs distribution

$$p_j^* = \frac{p_j^0 \exp\left(\frac{u_j}{\tau}\right)}{\sum_{k=1}^n p_k^0 \exp\left(\frac{u_k}{\tau}\right)}.$$

Proof. The Lagrangian is

$$\mathcal{L}(p,\lambda) = \sum_{j=1}^n p_j u_j - \tau \sum_{j=1}^n p_j \log \frac{p_j}{p_j^0} + \lambda(1 - \sum_{j=1}^n p_j).$$

Differentiating with respect to p_j and setting it to zero, we have

$$\frac{\partial \mathcal{L}}{\partial p_j} = u_j - \tau \log \frac{p_j}{p_j^0} - \tau - \lambda = 0.$$

Rearranging, we have

$$p_j = p_j^0 \exp\left(\frac{u_j - \tau - \lambda}{\tau}\right) = p_j^0 \exp\left(\frac{u_j}{\tau}\right) \exp\left(-\frac{\tau + \lambda}{\tau}\right).$$

Normalization gives

$$p_j^* = \frac{p_j^0 \exp\left(\frac{u_j}{\tau}\right)}{\sum_{k=1}^n p_k^0 \exp\left(\frac{u_k}{\tau}\right)}.$$

To prove global optimality, define $Z = \sum_{k=1}^n p_k^0 \exp\left(\frac{u_k}{\tau}\right)$. Since

$$\log p_j^* = \log p_j^0 + \frac{u_j}{\tau} - \log Z,$$

we obtain, for every $p \in \Delta_n$,

$$\tau \mathrm{KL}(p \| p^*) = \tau \sum_{j=1}^n p_j \log \frac{p_j}{p_j^*} = \tau \sum_{j=1}^n p_j \log \frac{p_j}{p_j^0} - \sum_{j=1}^n p_j u_j + \tau \log Z.$$

Thus, we have

$$F(p) = -\tau \mathrm{KL}(p\|p^*) + \tau \log Z.$$

Since KL divergence is non-negative, we have $F(p) \le \tau \log Z = F(p^*)$ for all $p \in \Delta_n$. Moreover, $\mathrm{KL}(p\|p^*) = 0$ if and only if $p = p^*$, so p^* is the unique global maximizer of $F(p)$. □

Assumption D.2. Assume that $\mathcal{U}, \mathcal{Q}, \mathcal{A}$ are finite nonempty sets, and that the following conditions hold:

1. f is continuously differentiable, C_f-Lipschitz continuous and L_f-smooth in Z.
2. The incentive reward model $r_{x,i}$ is bounded by B_r, i.e., $|r_{x,i}(s,q,a)| \le B_r$ for all $x \in \mathbb{R}^{d_x}$, s, q, a and $i \in \{Q, S\}$.
3. The incentive reward model $r_{x,i}$ is continuously differentiable, C_r-Lipschitz continuous and L_r-smooth in $\mathbb{R}^{d_x}$ for $i \in \{Q, S\}$.
4. For every $s \in \mathcal{U}$ and $q \in \mathcal{Q}$, the scalar map $\phi \mapsto \pi_\phi(q|s)$ is continuously differentiable. There exist $C_\phi, L_\phi < \infty$ such that, uniformly over s, q and all parameter pairs,
$$\|\pi_{\phi_1}(\cdot|s) - \pi_{\phi_2}(\cdot|s)\|_1 \le C_\phi \|\phi_1 - \phi_2\|_2,$$
$$\|\nabla_\phi \pi_{\phi_1}(q|s) - \nabla_\phi \pi_{\phi_2}(q|s)\|_2 \le L_\phi \|\phi_1 - \phi_2\|_2.$$
5. For every $q \in \mathcal{Q}$ and $a \in \mathcal{A}$, the scalar map $\psi \mapsto \pi_\psi(a|q)$ is continuously differentiable. There exist $C_\psi, L_\psi < \infty$ such that, uniformly over q, a and all parameter pairs,
$$\|\pi_{\psi_1}(\cdot|q) - \pi_{\psi_2}(\cdot|q)\|_1 \le C_\psi \|\psi_1 - \psi_2\|_2,$$
$$\|\nabla_\psi \pi_{\psi_1}(a|q) - \nabla_\psi \pi_{\psi_2}(a|q)\|_2 \le L_\psi \|\psi_1 - \psi_2\|_2.$$
6. The regularization strengths $\tau_Q, \tau_S > 0$ are fixed. The reference policies are fixed independently of (x, ϕ, ψ) and have full support. In particular,
$$\eta_Q \triangleq \min_{s,q} \pi^0_\phi(q|s) > 0, \qquad \eta_S \triangleq \min_{q,a} \pi^0_\psi(a|q) > 0.$$
7. The parameterization is sufficiently expressive so that for every x and fixed opponent policy, each player's policy class admits a parameter vector that simultaneously realizes all unrestricted conditional Gibbs best responses.
8. The distribution ν is a fixed probability distribution on $\mathcal{U}$, independent of (x, ϕ, ψ), with full support: $\nu(s) > 0$ for all $s \in \mathcal{U}$ and $\sum_s \nu(s) = 1$.
9. Suppose π_ϕ and π_ψ have full support. Moreover, there exist $\delta_Q > 0$ and $\delta_S > 0$ such that $\pi_\phi(q|s) \ge \delta_Q$ for all $\phi \in \mathbb{R}^{d_\phi}$, $q \in \mathcal{Q}$ and $s \in \mathcal{U}$, and $\pi_\psi(a|q) \ge \delta_S$ for all $\psi \in \mathbb{R}^{d_\psi}$, $a \in \mathcal{A}$ and $q \in \mathcal{Q}$.

Lemma D.3. *Under Assumption D.2, the LL objective functions $g_Q(x, \phi, \psi)$ and $g_S(x, \phi, \psi)$ are continuously differentiable, C_Q- and C_S-Lipschitz continuous, and L_Q- and L_S-smooth in Z, respectively.*

Proof. The differentiability of g_Q and g_S follows from the differentiability of the policy functions and the reward functions by Assumption D.2.

Let

$$M = \max\{\log(1/\delta_Q), \log(1/\delta_S), \log(1/\eta_Q), \log(1/\eta_S)\}.$$

Then

$$\left|\log \frac{\pi_\phi(q|s)}{\pi^0_\phi(q|s)}\right| \le M, \qquad \left|\log \frac{\pi_\psi(a|q)}{\pi^0_\psi(a|q)}\right| \le M, \qquad \|\nabla_x r_{x,i}(s,q,a)\| \le C_r.$$

Moreover, for any $v \in \mathbb{R}^{|\mathcal{Q}|}$,

$$\left\|\sum_q v_q \nabla_\phi \pi_\phi(q|s)\right\| = \sup_{\|h\|=1} \left|\lim_{t\to 0} \sum_q v_q \frac{\pi_{\phi+th}(q|s) - \pi_\phi(q|s)}{t}\right| \le C_\phi \|v\|_\infty,$$

and, similarly, for any $v \in \mathbb{R}^{|\mathcal{A}|}$,

$$\left\|\sum_a v_a \nabla_\psi \pi_\psi(a|q)\right\| \le C_\psi \|v\|_\infty.$$

Since $\sum_q \nabla_\phi \pi_\phi(q|s) = 0$ and $\sum_a \nabla_\psi \pi_\psi(a|q) = 0$, we have

$$\nabla_x g_i(z) = -\sum_{s,q,a} \nu(s)\pi_\phi(q|s)\pi_\psi(a|q)\nabla_x r_{x,i}(s,q,a), \qquad i \in \{Q,S\},$$

$$\nabla_\phi g_Q(z) = \sum_{s,q} \nu(s)\nabla_\phi \pi_\phi(q|s)\left[\tau_Q \log\frac{\pi_\phi(q|s)}{\pi_\phi^0(q|s)} - \sum_a \pi_\psi(a|q) r_{x,Q}(s,q,a)\right],$$

$$\nabla_\psi g_Q(z) = -\sum_{s,q,a} \nu(s)\pi_\phi(q|s) r_{x,Q}(s,q,a)\nabla_\psi \pi_\psi(a|q),$$

$$\nabla_\phi g_S(z) = \sum_{s,q,a} \nu(s)\pi_\psi(a|q)\nabla_\phi \pi_\phi(q|s)\left[\tau_S \log\frac{\pi_\psi(a|q)}{\pi_\psi^0(a|q)} - r_{x,S}(s,q,a)\right],$$

$$\nabla_\psi g_S(z) = \sum_{s,q,a} \nu(s)\pi_\phi(q|s)\nabla_\psi \pi_\psi(a|q)\left[\tau_S \log\frac{\pi_\psi(a|q)}{\pi_\psi^0(a|q)} - r_{x,S}(s,q,a)\right].$$

Consequently,

$$\|\nabla_x g_i(z)\| \le C_r, \qquad i \in \{Q,S\},$$
$$\|\nabla_\phi g_Q(z)\| \le C_\phi(B_r + \tau_Q M), \qquad \|\nabla_\psi g_Q(z)\| \le B_r C_\psi,$$
$$\|\nabla_\phi g_S(z)\| \le C_\phi(B_r + \tau_S M), \qquad \|\nabla_\psi g_S(z)\| \le C_\psi(B_r + \tau_S M).$$

Taking

$$C_Q = C_r + C_\phi(B_r + \tau_Q M) + B_r C_\psi, \qquad C_S = C_r + (C_\phi + C_\psi)(B_r + \tau_S M),$$

gives $\|\nabla g_i(z)\| \le C_i$ and

$$|g_i(z_1) - g_i(z_2)| = \left|\int_0^1 \langle \nabla g_i(z_2 + t(z_1 - z_2)), z_1 - z_2\rangle \, dt\right| \le C_i\|z_1 - z_2\|, \qquad i \in \{Q,S\}.$$

For smoothness, the policy lower bounds imply

$$\max_{s,q}\left|\log\frac{\pi_{\phi_1}(q|s)}{\pi_{\phi_2}(q|s)}\right| \le \frac{C_\phi}{\delta_Q}\|\phi_1 - \phi_2\|,$$
$$\max_{q,a}\left|\log\frac{\pi_{\psi_1}(a|q)}{\pi_{\psi_2}(a|q)}\right| \le \frac{C_\psi}{\delta_S}\|\psi_1 - \psi_2\|.$$

Subtracting the formulas for $\nabla_x g_i$ and $\nabla_\phi g_Q$ yields

$$\|\nabla_x g_i(z_1) - \nabla_x g_i(z_2)\| \le L_r\|x_1 - x_2\| + C_r C_\phi\|\phi_1 - \phi_2\| + C_r C_\psi\|\psi_1 - \psi_2\|.$$

$$\begin{aligned}
&\|\nabla_\phi g_Q(z_1) - \nabla_\phi g_Q(z_2)\| \\
&\le |\mathcal{Q}| L_\phi(B_r + \tau_Q M)\|\phi_1 - \phi_2\| \\
&\quad + C_\phi \max_{s,q}\left|\tau_Q \log\frac{\pi_{\phi_1}(q|s)}{\pi_{\phi_2}(q|s)} - \sum_a \left[\pi_{\psi_1}(a|q) r_{x_1,Q}(s,q,a) - \pi_{\psi_2}(a|q) r_{x_2,Q}(s,q,a)\right]\right| \\
&\le C_r C_\phi\|x_1 - x_2\| + \left[|\mathcal{Q}| L_\phi(B_r + \tau_Q M) + \frac{\tau_Q C_\phi^2}{\delta_Q}\right]\|\phi_1 - \phi_2\| \\
&\quad + B_r C_\phi C_\psi\|\psi_1 - \psi_2\|.
\end{aligned}$$

Similarly,

$$\begin{aligned}\|\nabla_\psi g_Q(z_1) - \nabla_\psi g_Q(z_2)\| &\le C_r C_\psi \|x_1 - x_2\| + B_r C_\phi C_\psi \|\phi_1 - \phi_2\| \\ &\quad + B_r |\mathcal{A}| L_\psi \|\psi_1 - \psi_2\|, \\ \|\nabla_\psi g_S(z_1) - \nabla_\psi g_S(z_2)\| &\le C_r C_\psi \|x_1 - x_2\| + C_\phi C_\psi (B_r + \tau_S M)\|\phi_1 - \phi_2\| \\ &\quad + \left[|\mathcal{A}| L_\psi (B_r + \tau_S M) + \frac{\tau_S C_\psi^2}{\delta_S} \right] \|\psi_1 - \psi_2\|.\end{aligned}$$

For the remaining block, normalization gives

$$\begin{aligned}&\left\| \nabla_\psi \sum_a \pi_\psi(a|q) \left[\tau_S \log \frac{\pi_\psi(a|q)}{\pi_\psi^0(a|q)} - r_{x,S}(s,q,a) \right] \right\| \\ &= \left\| \sum_a \nabla_\psi \pi_\psi(a|q) \left[\tau_S \log \frac{\pi_\psi(a|q)}{\pi_\psi^0(a|q)} - r_{x,S}(s,q,a) \right] \right\| \le C_\psi (B_r + \tau_S M).\end{aligned}$$

Hence,

$$\begin{aligned}\|\nabla_\phi g_S(z_1) - \nabla_\phi g_S(z_2)\| &\le C_r C_\phi \|x_1 - x_2\| + |\mathcal{Q}| L_\phi (B_r + \tau_S M)\|\phi_1 - \phi_2\| \\ &\quad + C_\phi C_\psi (B_r + \tau_S M)\|\psi_1 - \psi_2\|.\end{aligned}$$

Combining these inequalities, we obtain

$$\begin{aligned}\|\nabla g_i(z_1) - \nabla g_i(z_2)\| &\le \sum_{b \in \{x,\phi,\psi\}} \|\nabla_b g_i(z_1) - \nabla_b g_i(z_2)\| \\ &\le L_i \|z_1 - z_2\|, \qquad i \in \{Q, S\},\end{aligned}$$

where

$$\begin{aligned}L_Q &= L_r + 2C_r(C_\phi + C_\psi) + B_r\big(|\mathcal{Q}| L_\phi + |\mathcal{A}| L_\psi + 2C_\phi C_\psi\big) \\ &\quad + \tau_Q \left(M |\mathcal{Q}| L_\phi + \frac{C_\phi^2}{\delta_Q} \right), \\ L_S &= L_r + 2C_r(C_\phi + C_\psi) + (B_r + \tau_S M)\big(|\mathcal{Q}| L_\phi + |\mathcal{A}| L_\psi + 2C_\phi C_\psi\big) + \frac{\tau_S C_\psi^2}{\delta_S}.\end{aligned}$$

□

Lemma D.4 (Best Response Policies)**.** *Under Assumption D.2, the best response policies $\pi_\phi^*(x,\psi)$ and $\pi_\psi^*(x,\phi)$ are given by*

$$\pi_\phi^*(q|s;x,\psi) = \frac{\pi_\phi^0(q|s) \exp\left(\frac{1}{\tau_Q} U_Q(x,\psi;s,q)\right)}{\sum_{q'} \pi_\phi^0(q'|s) \exp\left(\frac{1}{\tau_Q} U_Q(x,\psi;s,q')\right)}, \tag{38}$$

$$\pi_\psi^*(a|q;x,\phi) = \frac{\pi_\psi^0(a|q) \exp\left(\frac{1}{\tau_S} U_S(x,\phi;q,a)\right)}{\sum_{a'} \pi_\psi^0(a'|q) \exp\left(\frac{1}{\tau_S} U_S(x,\phi;q,a')\right)}, \tag{39}$$

where

$$\begin{aligned}U_Q(x,\psi;s,q) &= \sum_a \pi_\psi(a|q) r_{x,Q}(s,q,a), \\ U_S(x,\phi;q,a) &= \frac{\sum_s \nu(s)\pi_\phi(q|s) r_{x,S}(s,q,a)}{\sum_s \nu(s)\pi_\phi(q|s)}.\end{aligned}$$

Proof. For the questioner, define the expected reward of proposing a question q under a seed prompt s as

$$U_Q(x,\psi;s,q) = \sum_{a \in \mathcal{A}} \pi_\psi(a|q) r_{x,Q}(s,q,a).$$

With x, ψ fixed, rewrite the questioner objective as

$$J_Q(x,\phi,\psi) = \sum_{s\in\mathcal{U}} \nu(s) \left[\sum_{q\in\mathcal{Q}} \pi_\phi(q|s) U_Q(x,\psi;s,q) - \tau_Q \sum_{q\in\mathcal{Q}} \pi_\phi(q|s) \log \frac{\pi_\phi(q|s)}{\pi^0_\phi(q|s)} \right].$$

Maximizing the questioner objective is equivalent to the following optimization problems,

$$\max_{p\in\Delta(\mathcal{Q})} \left\{ \sum_{q\in\mathcal{Q}} p(q) U_Q(x,\psi;s,q) - \tau_Q \mathrm{KL}(p\|\pi^0_\phi(\cdot|s)) \right\}, \tag{40}$$

for each s with $\nu(s) > 0$.

By Lemma D.1, the unique global maximizer is given by

$$\pi^*_\phi(q|s;x,\psi) = \frac{\pi^0_\phi(q|s) \exp\left(\frac{U_Q(x,\psi;s,q)}{\tau_Q}\right)}{\sum_{q'\in\mathcal{Q}} \pi^0_\phi(q'|s) \exp\left(\frac{U_Q(x,\psi;s,q')}{\tau_Q}\right)}.$$

For the solver, define the corresponding expected reward as

$$U_S(x,\phi;q,a) = \sum_{s\in\mathcal{U}} \frac{\nu(s)\pi_\phi(q|s)}{\sum_{s\in\mathcal{U}} \nu(s)\pi_\phi(q|s)} r_{x,S}(s,q,a).$$

Now with x, ϕ fixed, rewrite the solver objective as

$$J_S(x,\phi,\psi) = \sum_{s\in\mathcal{U}} \sum_{q\in\mathcal{Q}} \nu(s)\pi_\phi(q|s) \left[\sum_{a\in\mathcal{A}} \pi_\psi(a|q) U_S(x,\phi;q,a) - \tau_S \sum_{a\in\mathcal{A}} \pi_\psi(a|q) \log \frac{\pi_\psi(a|q)}{\pi^0_\psi(a|q)} \right].$$

For every question q with $\sum_{s\in\mathcal{U}} \nu(s)\pi_\phi(q|s) > 0$, the conditional optimization problem is

$$\max_{p\in\Delta(\mathcal{A})} \left\{ \sum_{a\in\mathcal{A}} p(a) U_S(x,\phi;q,a) - \tau_S \mathrm{KL}(p\|\pi^0_\psi(\cdot|q)) \right\}. \tag{41}$$

By Lemma D.1, the unique global maximizer is given by

$$\pi^*_\psi(a|q;x,\phi) = \frac{\pi^0_\psi(a|q) \exp\left(\frac{U_S(x,\phi;q,a)}{\tau_S}\right)}{\sum_{a'\in\mathcal{A}} \pi^0_\psi(a'|q) \exp\left(\frac{U_S(x,\phi;q,a')}{\tau_S}\right)}.$$

□

Lemma D.5 (Lipschitz Continuity of the x-Partial Envelope Gradients)**.** *Under Assumption D.2, the best response envelopes are differentiable with respect to x. The maps $(x,\psi) \mapsto \nabla_x g^*_Q(x,\psi)$ and $(x,\phi) \mapsto \nabla_x g^*_S(x,\phi)$ are jointly Lipschitz continuous, with constants $L_{Q,x,*}$ and $L_{S,x,*}$, respectively.*

Proof. Incorporating the best response policies from Lemma D.4, we can rewrite the best response envelope as

$$g^*_Q(x,\psi) = -\tau_Q \sum_{s\in\mathcal{U}} \nu(s) \log \sum_{q\in\mathcal{Q}} \pi^0_\phi(q|s) \exp\left(\frac{U_Q(x,\psi;s,q)}{\tau_Q}\right), \tag{42}$$

$$g^*_S(x,\phi) = -\tau_S \sum_{q\in\mathcal{Q}} d_\phi(q) \log \sum_{a\in\mathcal{A}} \pi^0_\psi(a|q) \exp\left(\frac{U_S(x,\phi;q,a)}{\tau_S}\right), \tag{43}$$

where $d_\phi(q) \triangleq \sum_{s\in\mathcal{U}} \nu(s)\pi_\phi(q|s)$ is the marginal distribution of questions under the questioner policy π_ϕ and the prompt distribution ν.

Taking the gradient with respect to x, we can get

$$\nabla_x g_Q^*(x,\psi) = -\sum_{s\in\mathcal{U}} \nu(s) \sum_{q\in\mathcal{Q}} \pi_\phi^*(q|s;x,\psi)\nabla_x U_Q(x,\psi;s,q) \tag{44}$$

$$\nabla_x g_S^*(x,\phi) = -\sum_{q\in\mathcal{Q}} d_\phi(q) \sum_{a\in\mathcal{A}} \pi_\psi^*(a|q;x,\phi)\nabla_x U_S(x,\phi;q,a). \tag{45}$$

For any $x_1, x_2 \in \mathbb{R}^{d_x}$, we have

$$\begin{aligned}
&\|\nabla_x g_Q^*(x_1,\psi) - \nabla_x g_Q^*(x_2,\psi)\| \\
&\le \sum_{s\in\mathcal{U}} \nu(s) \| \sum_{q\in\mathcal{Q}} \pi_\phi^*(q|s;x_1,\psi)\nabla_x U_Q(x_1,\psi;s,q) - \sum_{q\in\mathcal{Q}} \pi_\phi^*(q|s;x_2,\psi)\nabla_x U_Q(x_2,\psi;s,q)\| \\
&\le \sum_{s\in\mathcal{U}} \nu(s) \|\pi_\phi^*(\cdot|s;x_1,\psi) - \pi_\phi^*(\cdot|s;x_2,\psi)\|_1 \max_{q\in\mathcal{Q}} \|\nabla_x U_Q(x_1,\psi;s,q)\| \\
&\quad + \sum_{s\in\mathcal{U}} \nu(s) \sum_{q\in\mathcal{Q}} \|\pi_\phi^*(q|s;x_2,\psi)\| \|\nabla_x U_Q(x_1,\psi;s,q) - \nabla_x U_Q(x_2,\psi;s,q)\|,
\end{aligned}$$

and

$$\begin{aligned}
&\|\nabla_x g_S^*(x_1,\phi) - \nabla_x g_S^*(x_2,\phi)\| \\
&\le \sum_{q\in\mathcal{Q}} d_\phi(q) \|\pi_\psi^*(\cdot|q;x_1,\phi) - \pi_\psi^*(\cdot|q;x_2,\phi)\|_1 \max_{a\in\mathcal{A}} \|\nabla_x U_S(x_1,\phi;q,a)\| \\
&\quad + \sum_{q\in\mathcal{Q}} d_\phi(q) \sum_{a\in\mathcal{A}} \|\pi_\psi^*(a|q;x_2,\phi)\| \|\nabla_x U_S(x_1,\phi;q,a) - \nabla_x U_S(x_2,\phi;q,a)\|.
\end{aligned}$$

Assumption D.2 implies that

$$\|\nabla_x U_Q(x_1,\psi;s,q)\| = \left\| \sum_{a\in\mathcal{A}} \pi_\psi(a|q)\ \nabla_x r_{x,Q}(s,q,a)|_{x=x_1} \right\| \le C_r,$$

$$\|\nabla_x U_S(x_1,\phi;q,a)\| = \left\| \sum_{s\in\mathcal{U}} \frac{\nu(s)\pi_\phi(q|s)}{\sum_{s'\in\mathcal{U}}\nu(s')\pi_\phi(q|s')}\ \nabla_x r_{x,S}(s,q,a)|_{x=x_1} \right\| \le C_r,$$

$$\|\nabla_x U_Q(x_1,\psi;s,q) - \nabla_x U_Q(x_2,\psi;s,q)\| \le L_r\|x_1-x_2\|,$$

$$\|\nabla_x U_S(x_1,\phi;q,a) - \nabla_x U_S(x_2,\phi;q,a)\| \le L_r\|x_1-x_2\|,$$

and

$$\begin{aligned}
\|\pi_\phi^*(\cdot|s;x_1,\psi) - \pi_\phi^*(\cdot|s;x_2,\psi)\|_1 &\le \frac{1}{\tau_Q}\|U_Q(x_1,\psi;s,\cdot) - U_Q(x_2,\psi;s,\cdot)\|_\infty \\
&\le \frac{C_r}{\tau_Q}\|x_1-x_2\|, \\
\|\pi_\psi^*(\cdot|q;x_1,\phi) - \pi_\psi^*(\cdot|q;x_2,\phi)\|_1 &\le \frac{1}{\tau_S}\|U_S(x_1,\phi;q,\cdot) - U_S(x_2,\phi;q,\cdot)\|_\infty \\
&\le \frac{C_r}{\tau_S}\|x_1-x_2\|.
\end{aligned}$$

Thus, we can conclude that

$$\|\nabla_x g_Q^*(x_1,\psi) - \nabla_x g_Q^*(x_2,\psi)\| \le \left(\frac{C_r^2}{\tau_Q} + L_r\right)\|x_1-x_2\| \tag{46}$$

$$\|\nabla_x g_S^*(x_1,\phi) - \nabla_x g_S^*(x_2,\phi)\| \le \left(\frac{C_r^2}{\tau_S} + L_r\right)\|x_1-x_2\| \tag{47}$$

For any $\psi_1, \psi_2 \in \mathbb{R}^{d_\psi}$, we have

$$\begin{aligned}
&\|\nabla_x g_Q^*(x,\psi_1) - \nabla_x g_Q^*(x,\psi_2)\| \\
\le &\sum_{s\in\mathcal{U}} \nu(s)\|\pi_\phi^*(\cdot|s;x,\psi_1) - \pi_\phi^*(\cdot|s;x,\psi_2)\|_1 \max_{q\in\mathcal{Q}} \|\nabla_x U_Q(x,\psi_1;s,q)\| \\
&+ \sum_{s\in\mathcal{U}} \nu(s) \sum_{q\in\mathcal{Q}} \|\pi_\phi^*(q|s;x,\psi_2)\|\|\nabla_x U_Q(x,\psi_1;s,q) - \nabla_x U_Q(x,\psi_2;s,q)\|.
\end{aligned}$$

Assumption D.2 implies that

$$\begin{aligned}
&\|\nabla_x U_Q(x,\psi_1;s,q) - \nabla_x U_Q(x,\psi_2;s,q)\| \\
= &\|\sum_{a\in\mathcal{A}} (\pi_{\psi_1}(a|q) - \pi_{\psi_2}(a|q))\nabla_x r_{x,Q}(s,q,a)\| \\
\le &C_r C_\psi \|\psi_1 - \psi_2\|,
\end{aligned}$$

and

$$\begin{aligned}
&\|\pi_\phi^*(\cdot|s;x,\psi_1) - \pi_\phi^*(\cdot|s;x,\psi_2)\|_1 \\
\le &\frac{1}{\tau_Q}\|U_Q(x,\psi_1;s,\cdot) - U_Q(x,\psi_2;s,\cdot)\|_\infty \\
= &\frac{1}{\tau_Q}\max_{q\in\mathcal{Q}}\left|\sum_{a\in\mathcal{A}} (\pi_{\psi_1}(a|q) - \pi_{\psi_2}(a|q)) r_{x,Q}(s,q,a)\right| \\
\le &\frac{B_r C_\psi}{\tau_Q}\|\psi_1 - \psi_2\|.
\end{aligned}$$

Consequently,

$$\|\nabla_x g_Q^*(x,\psi_1) - \nabla_x g_Q^*(x,\psi_2)\| \le C_r C_\psi \left(\frac{B_r}{\tau_Q} + 1\right)\|\psi_1 - \psi_2\|. \tag{48}$$

Now fix x and consider variation of $\nabla_x g_S^*(x,\phi)$ in ϕ. For $j \in \{1,2\}$, define

$$\begin{gathered}
w_j(s,q) = \nu(s)\pi_{\phi_j}(q|s), \qquad d_j(q) = \sum_s w_j(s,q), \qquad \Delta_q = \sum_s |w_1(s,q) - w_2(s,q)|, \\
U_j(q,a) = U_S(x,\phi_j;q,a), \qquad p_j(a|q) = \pi_\psi^*(a|q;x,\phi_j), \\
b_j(q,a) = \sum_s w_j(s,q)\nabla_x r_{x,S}(s,q,a).
\end{gathered}$$

Then

$$\begin{gathered}
|d_1(q) - d_2(q)| \le \Delta_q, \qquad \sum_q \Delta_q = \sum_s \nu(s)\|\pi_{\phi_1}(\cdot|s) - \pi_{\phi_2}(\cdot|s)\|_1 \le C_\phi\|\phi_1 - \phi_2\|, \\
\max_a \|b_1(q,a) - b_2(q,a)\| \le C_r\Delta_q, \qquad \max_a \|b_2(q,a)\| \le C_r d_2(q).
\end{gathered}$$

Since $d_j(q) > 0$ and $\|U_j(q,\cdot)\|_\infty \le B_r$, we have

$$\begin{aligned}
d_2(q)\big(U_1(q,\cdot) - U_2(q,\cdot)\big) &= \sum_s \big(w_1(s,q) - w_2(s,q)\big) r_{x,S}(s,q,\cdot) \\
&\quad + \big(d_2(q) - d_1(q)\big)U_1(q,\cdot), \\
d_2(q)\|U_1(q,\cdot) - U_2(q,\cdot)\|_\infty &\le B_r\Delta_q + B_r|d_2(q) - d_1(q)| \\
&\le 2B_r\Delta_q.
\end{aligned}$$

The envelope gradients can be written as

$$\nabla_x g_S^*(x,\phi_j) = -\sum_q \sum_a p_j(a|q) b_j(q,a).$$

Consequently,

$$\begin{aligned}
&\left\| \sum_a p_1(a|q) b_1(q,a) - \sum_a p_2(a|q) b_2(q,a) \right\| \\
&\leq \sum_a p_1(a|q) \|b_1(q,a) - b_2(q,a)\| + \|p_1(\cdot|q) - p_2(\cdot|q)\|_1 \max_a \|b_2(q,a)\| \\
&\leq C_r \Delta_q + \frac{C_r d_2(q)}{\tau_S} \|U_1(q,\cdot) - U_2(q,\cdot)\|_\infty \\
&\leq C_r \left(1 + \frac{2B_r}{\tau_S}\right) \Delta_q.
\end{aligned}$$

Summing over q gives

$$\begin{aligned}
\|\nabla_x g_S^*(x,\phi_1) - \nabla_x g_S^*(x,\phi_2)\| &\leq C_r \left(1 + \frac{2B_r}{\tau_S}\right) \sum_q \Delta_q \\
&\leq C_r C_\phi \left(1 + \frac{2B_r}{\tau_S}\right) \|\phi_1 - \phi_2\|. \qquad (49)
\end{aligned}$$

To combine the separate-block estimates, define

$$\begin{aligned}
A_Q &= L_r + \frac{C_r^2}{\tau_Q}, & B_Q &= C_r C_\psi \left(1 + \frac{B_r}{\tau_Q}\right), \\
A_S &= L_r + \frac{C_r^2}{\tau_S}, & B_S &= C_r C_\phi \left(1 + \frac{2B_r}{\tau_S}\right).
\end{aligned}$$

Inserting an intermediate point that changes only x yields

$$\begin{aligned}
\|\nabla_x g_Q^*(x_1,\psi_1) - \nabla_x g_Q^*(x_2,\psi_2)\| &\leq A_Q \|x_1 - x_2\| + B_Q \|\psi_1 - \psi_2\|, \\
\|\nabla_x g_S^*(x_1,\phi_1) - \nabla_x g_S^*(x_2,\phi_2)\| &\leq A_S \|x_1 - x_2\| + B_S \|\phi_1 - \phi_2\|.
\end{aligned}$$

By Cauchy–Schwarz, these imply

$$\|\nabla_x g_Q^*(x_1,\psi_1) - \nabla_x g_Q^*(x_2,\psi_2)\| \leq L_{Q,x,*} \|(x_1,\psi_1) - (x_2,\psi_2)\|, \qquad (50)$$

$$\|\nabla_x g_S^*(x_1,\phi_1) - \nabla_x g_S^*(x_2,\phi_2)\| \leq L_{S,x,*} \|(x_1,\phi_1) - (x_2,\phi_2)\|, \qquad (51)$$

where

$$L_{Q,x,*} = \sqrt{A_Q^2 + B_Q^2}, \qquad L_{S,x,*} = \sqrt{A_S^2 + B_S^2}.$$

□

Lemma D.6. *Under Assumption D.2, for fixed $\lambda > 0$, the operator $\mathcal{H}_\lambda$ is L_H-Lipschitz continuous with*

$$L_H = L_f + \lambda\big(L_Q + L_S + L_{Q,x,*} + L_{S,x,*}\big).$$

In particular, for any $z_1, z_2 \in Z$,

$$\|\mathcal{H}_\lambda(z_1) - \mathcal{H}_\lambda(z_2)\| \leq L_H \|z_1 - z_2\|.$$

Proof. Let Π_Q retain the x, ϕ blocks of a vector and set its ψ block to zero, and let Π_S retain the x, ψ blocks and set its ϕ block to zero. Let $E_x v = (v, 0, 0)$ embed an x-vector into Z. By the definition of the mixed operator,

$$\begin{aligned}
\mathcal{H}_\lambda(z) = \nabla f(z) &+ \lambda \Pi_Q \nabla g_Q(z) + \lambda \Pi_S \nabla g_S(z) \\
&- \lambda E_x \big(\nabla_x g_Q^*(x,\psi) + \nabla_x g_S^*(x,\phi)\big).
\end{aligned}$$

The projections have operator norm at most one, and E_x preserves the Euclidean norm. Assumption D.2 and Lemmas D.3 and D.5 therefore give, by the triangle inequality,

$$\begin{aligned}
\|\mathcal{H}_\lambda(z_1) - \mathcal{H}_\lambda(z_2)\| &\leq \big[L_f + \lambda(L_Q + L_S + L_{Q,x,*} + L_{S,x,*})\big] \|z_1 - z_2\| \\
&= L_H \|z_1 - z_2\|.
\end{aligned}$$

□

Assumption D.7 (Weak Minty Condition)**.** There are $z^* \in Z$ and $\kappa \geq 0$ such that $\mathcal{H}_\lambda(z^*) = 0$ and

$$\langle \mathcal{H}_\lambda(z), z - z^* \rangle \geq -\kappa \|\mathcal{H}_\lambda(z)\|^2, \quad \forall z \in Z.$$

For fixed $\rho > 0$, write

$$\mathcal{F}_\lambda(z, w) = \mathcal{H}_\lambda(z) + \rho(z - w).$$

The exact-oracle iteration is

$$z_{t+1} = z_t - \eta \mathcal{F}_\lambda(z_t, w_t), \qquad w_{t+1} = w_t + \beta(z_{t+1} - w_t). \tag{52}$$

Theorem D.8 (Convergence of the exact-oracle algorithm)**.** *Under Assumptions D.2 and D.7, suppose $L_H > 0$, $\kappa L_H < 1$, where L_H is the Lipschitz constant of $\mathcal{H}_\lambda$ provided in Lemma D.6, and best responses are computed exactly. Choose*

$$\rho = \frac{2L_H}{1 + \kappa L_H}, \qquad \eta = \frac{1 + \kappa L_H}{2L_H}, \qquad \beta = \frac{(1 - \kappa L_H)^2}{16(1 + \kappa L_H)}. \tag{53}$$

Then iterations generated by the Algorithm 2 satisfy, for every $T \geq 1$,

$$\frac{1}{T} \sum_{t=0}^{T-1} \|\mathcal{H}_\lambda(z_t)\|^2 \leq \frac{128 L_H^2 \|z_0 - z^*\|^2}{(1 - \kappa L_H)^3 T}.$$

Proof. Write $\mathcal{F}_{\lambda,t} = \mathcal{F}_\lambda(z_t, w_t)$. Since

$$z_{t+1} - w_{t+1} = (1 - \beta)(z_{t+1} - w_t),$$

and

$$\mathcal{F}_{\lambda,t} = \mathcal{H}_\lambda(z_t) + \rho(z_t - w_t),$$

direct substitution gives

$$\begin{aligned}
\mathcal{F}_{\lambda,t+1} &= \mathcal{H}_\lambda(z_{t+1}) + \rho(z_{t+1} - w_{t+1}) \\
&= \mathcal{H}_\lambda(z_{t+1}) + \rho(1 - \beta)(z_{t+1} - w_t) \\
&= \mathcal{H}_\lambda(z_{t+1}) + \rho(1 - \beta)\left(z_t - \eta \mathcal{F}_{\lambda,t} - w_t\right) \\
&= \mathcal{H}_\lambda(z_{t+1}) + \rho(1 - \beta)\left(\frac{1}{\rho}\mathcal{F}_{\lambda,t} - \frac{1}{\rho}\mathcal{H}_\lambda(z_t) - \eta \mathcal{F}_{\lambda,t}\right) \\
&= \mathcal{H}_\lambda(z_{t+1}) + (1 - \beta)(1 - \rho\eta)\mathcal{F}_{\lambda,t} - (1 - \beta)\mathcal{H}_\lambda(z_t).
\end{aligned} \tag{54}$$

By L_H-Lipschitz continuity,

$$\begin{aligned}
\|\mathcal{H}_\lambda(z_{t+1}) - \mathcal{H}_\lambda(z_t)\| &\leq L_H \|z_{t+1} - z_t\| \\
&= L_H \eta \|\mathcal{F}_{\lambda,t}\|.
\end{aligned}$$

Substituting this into equation 54 gives

$$\begin{aligned}
\|\mathcal{F}_{\lambda,t+1}\| &\leq L_H \eta \|\mathcal{F}_{\lambda,t}\| + (1 - \beta)|1 - \rho\eta| \|\mathcal{F}_{\lambda,t}\| + \beta \|\mathcal{H}_\lambda(z_t)\| \\
&= q \|\mathcal{F}_{\lambda,t}\| + \beta \|\mathcal{H}_\lambda(z_t)\|,
\end{aligned}$$

where $q \triangleq L_H \eta + (1 - \beta)|1 - \rho\eta|$.

Require

$$q = L_H \eta + (1 - \beta)|1 - \rho\eta| < 1, \tag{55}$$

Using the inequality

$$(qa + b)^2 \leq qa^2 + \frac{b^2}{1 - q}, \quad 0 \leq q < 1,$$

yields

$$\|\mathcal{F}_{\lambda,t+1}\|^2 - \|\mathcal{F}_{\lambda,t}\|^2 \leq -(1-q)\|\mathcal{F}_{\lambda,t}\|^2 + \frac{\beta^2}{1-q}\|\mathcal{H}_\lambda(z_t)\|^2. \tag{56}$$

Consider $z_t - c(z_t - w_t)$ leaving c unspecified. Its change is

$$\begin{aligned}
&[z_{t+1} - c(z_{t+1} - w_{t+1})] - [z_t - c(z_t - w_t)] \\
&= (1 - c(1-\beta))\, z_{t+1} + c(1-\beta)w_t - z_t + c(z_t - w_t) \\
&= (1 - c(1-\beta))\, (z_t - \eta\mathcal{H}_\lambda(z_t) - \eta\rho(z_t - w_t)) + c(1-\beta)w_t - z_t + c(z_t - w_t) \\
&= ((1 - c(1-\beta))\,(1 - \eta\rho) - 1 + c)\, z_t \\
&\quad + ((1 - c(1-\beta))\,\eta\rho + c(1-\beta) - c)\, w_t - (1 - c(1-\beta))\,\eta\mathcal{H}_\lambda(z_t) \\
&= (c\,(\beta + (1-\beta)\eta\rho) - \eta\rho)\,(z_t - w_t) - (1 - c(1-\beta))\,\eta\mathcal{H}_\lambda(z_t).
\end{aligned}$$

Choose c such that

$$c\,(\beta + (1-\beta)\eta\rho) - \eta\rho = 0,$$

which gives

$$c = \frac{\eta\rho}{\beta + (1-\beta)\eta\rho}. \tag{57}$$

Define $u_t \triangleq z_t - c(z_t - w_t)$. Then

$$\begin{aligned}
u_{t+1} &= u_t - (1 - c(1-\beta))\,\eta\mathcal{H}_\lambda(z_t) \\
&= u_t - \frac{c\beta}{\rho}\mathcal{H}_\lambda(z_t).
\end{aligned}$$

Define

$$V_t \triangleq \|u_t - z^*\|^2,$$

so that

$$\begin{aligned}
V_{t+1} - V_t &= \|u_{t+1} - z^*\|^2 - \|u_t - z^*\|^2 \\
&= \|u_t - z^* - \frac{c\beta}{\rho}\mathcal{H}_\lambda(z_t)\|^2 - \|u_t - z^*\|^2 \\
&= \frac{c^2\beta^2}{\rho^2}\|\mathcal{H}_\lambda(z_t)\|^2 - 2\frac{c\beta}{\rho}\langle u_t - z^*, \mathcal{H}_\lambda(z_t)\rangle.
\end{aligned} \tag{58}$$

Since

$$\begin{aligned}
u_t - z_t &= c(w_t - z_t) \\
&= \frac{c}{\rho}\left(\mathcal{H}_\lambda(z_t) - \mathcal{F}_{\lambda,t}\right),
\end{aligned}$$

substituting this into equation 58 gives

$$\begin{aligned}
V_{t+1} - V_t &= \frac{c^2\beta^2}{\rho^2}\|\mathcal{H}_\lambda(z_t)\|^2 - 2\frac{c\beta}{\rho}\langle u_t - z_t, \mathcal{H}_\lambda(z_t)\rangle - 2\frac{c\beta}{\rho}\langle z_t - z^*, \mathcal{H}_\lambda(z_t)\rangle \\
&= \frac{c^2\beta^2}{\rho^2}\|\mathcal{H}_\lambda(z_t)\|^2 - 2\frac{c^2\beta}{\rho^2}\langle \mathcal{H}_\lambda(z_t) - \mathcal{F}_{\lambda,t}, \mathcal{H}_\lambda(z_t)\rangle - 2\frac{c\beta}{\rho}\langle z_t - z^*, \mathcal{H}_\lambda(z_t)\rangle \\
&= \frac{c^2\beta}{\rho^2}\,(\beta - 2)\,\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta}{\rho^2}\langle \mathcal{F}_{\lambda,t}, \mathcal{H}_\lambda(z_t)\rangle - 2\frac{c\beta}{\rho}\langle z_t - z^*, \mathcal{H}_\lambda(z_t)\rangle \\
&\leq \frac{c^2\beta}{\rho^2}\,(\beta - 2)\,\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta}{\rho^2}\langle \mathcal{F}_{\lambda,t}, \mathcal{H}_\lambda(z_t)\rangle + 2\frac{c\beta}{\rho}\kappa\|\mathcal{H}_\lambda(z_t)\|^2 \\
&= \frac{c\beta}{\rho^2}\,(c(\beta - 2) + 2\kappa\rho)\,\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta}{\rho^2}\langle \mathcal{F}_{\lambda,t}, \mathcal{H}_\lambda(z_t)\rangle \\
&\leq \frac{c\beta}{\rho^2}\,(c(\beta - 2) + 2\kappa\rho)\,\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta}{\rho^2}\|\mathcal{F}_{\lambda,t}\|\|\mathcal{H}_\lambda(z_t)\|,
\end{aligned}$$

by the weak Minty condition and Cauchy–Schwarz inequality.

Young's inequality gives

$$2\beta\|\mathcal{F}_{\lambda,t}\|\|\mathcal{H}_\lambda(z_t)\| \le (1-q)\|\mathcal{F}_{\lambda,t}\|^2 + \frac{\beta^2}{1-q}\|\mathcal{H}_\lambda(z_t)\|^2.$$

Define the energy

$$E_t \triangleq V_t + \frac{c^2}{\rho^2}\|\mathcal{F}_{\lambda,t}\|^2. \tag{59}$$

Combining with equation 56,

$$\begin{aligned}
&E_{t+1} - E_t \\
&= V_{t+1} - V_t + \frac{c^2}{\rho^2}\left(\|\mathcal{F}_{\lambda,t+1}\|^2 - \|\mathcal{F}_{\lambda,t}\|^2\right) \\
&\le \frac{c\beta}{\rho^2}\left(c(\beta-2) + 2\kappa\rho\right)\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta}{\rho^2}\|\mathcal{F}_{\lambda,t}\|\|\mathcal{H}_\lambda(z_t)\| \\
&\quad - (1-q)\frac{c^2}{\rho^2}\|\mathcal{F}_{\lambda,t}\|^2 + \frac{c^2\beta^2}{(1-q)\rho^2}\|\mathcal{H}_\lambda(z_t)\|^2 \\
&\le \frac{c\beta}{\rho^2}\left(c(\beta-2) + 2\kappa\rho\right)\|\mathcal{H}_\lambda(z_t)\|^2 + \frac{2c^2\beta^2}{(1-q)\rho^2}\|\mathcal{H}_\lambda(z_t)\|^2 \\
&= \frac{c\beta}{\rho^2}\left(c(\beta-2) + 2\kappa\rho + \frac{2c\beta}{1-q}\right)\|\mathcal{H}_\lambda(z_t)\|^2.
\end{aligned}$$

It suffices to choose η, ρ, β such that

$$\begin{aligned}
q = L_H\eta + (1-\beta)|1-\rho\eta| < 1, \\
c(2-\beta) - 2\kappa\rho - \frac{2c\beta}{1-q} > 0, \\
c = \frac{\eta\rho}{\beta + (1-\beta)\eta\rho}.
\end{aligned}$$

Choose

$$\begin{cases}
L_H < \rho, \qquad \kappa\rho < 1, \\
0 < \beta \le \frac{(1-\kappa\rho)(1-\frac{L_H}{\rho})}{8}, \\
|\rho\eta - 1| \le \frac{1-\frac{L_H}{\rho}}{4}.
\end{cases} \tag{60}$$

These choices exist whenever $L_H\kappa < 1$, including $\kappa = 0$. Since $0 < \beta \le 1/8$ and $3/4 \le \rho\eta \le 5/4$,

$$|c-1| = \frac{\beta|\rho\eta - 1|}{\beta + (1-\beta)\rho\eta} \le \frac{\beta}{3}, \qquad 1-\beta \le c \le 1 + \frac{\beta}{3} < \frac{5}{4},$$

$$q \le \frac{L_H}{\rho} + \left(1 + \frac{L_H}{\rho}\right)|\rho\eta - 1| \le \frac{L_H}{\rho} + \frac{1-(L_H/\rho)^2}{4} \le \frac{1 + L_H/\rho}{2}.$$

Therefore,

$$q \le \frac{1 + \frac{L_H}{\rho}}{2} < 1,$$

and

$$\begin{aligned}
&c(2-\beta) - 2\kappa\rho - \frac{2c\beta}{1-q} \\
&\ge 2(1-\kappa\rho) - 3\beta - \frac{5\beta}{1-\frac{L_H}{\rho}} \\
&\ge 2(1-\kappa\rho) - \frac{8\beta}{1-\frac{L_H}{\rho}} \\
&\ge 1 - \kappa\rho > 0.
\end{aligned}$$

Therefore, defining $\alpha \triangleq \frac{c\beta(1-\kappa\rho)}{\rho^2} > 0$ gives

$$E_{t+1} - E_t \le -\alpha \|\mathcal{H}_\lambda(z_t)\|^2.$$

Since $w_0 = z_0$ and $\beta < 1 - L_H/\rho$,

$$\frac{cL_H}{\rho} \le \frac{L_H}{\rho(1-\beta)} < 1,$$

$$E_0 = \|z_0 - z^*\|^2 + \frac{c^2}{\rho^2}\|\mathcal{H}_\lambda(z_0)\|^2 \le \left(1 + \frac{c^2 L_H^2}{\rho^2}\right)\|z_0 - z^*\|^2 \le 2\|z_0 - z^*\|^2.$$

Telescoping yields

$$\alpha \sum_{t=0}^{T-1} \|\mathcal{H}_\lambda(z_t)\|^2 \le E_0 - E_T \le 2\|z_0 - z^*\|^2,$$

$$\min_{0\le t<T} \|\mathcal{H}_\lambda(z_t)\|^2 \le \frac{1}{T}\sum_{t=0}^{T-1} \|\mathcal{H}_\lambda(z_t)\|^2 \le \frac{2\|z_0 - z^*\|^2}{\alpha T}. \tag{61}$$

In particular,

$$\sum_{t=0}^{\infty} \|\mathcal{H}_\lambda(z_t)\|^2 \le \frac{2\|z_0 - z^*\|^2}{\alpha} < \infty, \qquad \|\mathcal{H}_\lambda(z_t)\| \longrightarrow 0.$$

If $\eta = 1/\rho$ and $0 < \beta \le (1-\kappa\rho)(1 - L_H/\rho)/4$, then

$$c = 1, \qquad q = L_H/\rho,$$

$$c(2-\beta) - 2\kappa\rho - \frac{2c\beta}{1-q} = 2(1-\kappa\rho) - \beta - \frac{2\beta}{1 - L_H/\rho}$$

$$\ge 2(1-\kappa\rho) - \frac{(1-\kappa\rho)(3 - L_H/\rho)}{4} \ge 1 - \kappa\rho.$$

Thus the same descent and telescoping inequalities hold. Using $\rho = 1/\eta$ and $w_0 = z_0$ gives

$$\alpha = \beta\eta^2(1 - \kappa/\eta), \qquad E_0 = \|z_0 - z^*\|^2 + \eta^2\|\mathcal{H}_\lambda(z_0)\|^2 \le 2\|z_0 - z^*\|^2.$$

For the choice equation 53,

$$\rho\eta = 1, \qquad 1 - \frac{L_H}{\rho} = \frac{1 - \kappa L_H}{2}, \qquad 1 - \kappa\rho = \frac{1 - \kappa L_H}{1 + \kappa L_H},$$

$$\beta = \frac{(1-\kappa\rho)(1 - L_H/\rho)}{8} > 0.$$

Hence equation 60 holds, and

$$c = 1, \qquad q = \frac{1 + \kappa L_H}{2}, \qquad \alpha = \frac{(1 - \kappa L_H)^3}{64 L_H^2}. \tag{62}$$

Substitution into equation 61 gives

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\mathcal{H}_\lambda(z_t)\|^2 \le \frac{128 L_H^2 \|z_0 - z^*\|^2}{(1 - \kappa L_H)^3 T}. \tag{63}$$

□

Assumption D.9 (LL Gradient Dominance)**.** There exist $\mu_Q, \mu_S > 0$ such that, at every analyzed iterate $z_t = (x_t, \phi_t, \psi_t)$,

$$g_Q(z_t) - g_Q^*(x_t, \psi_t) \le \frac{\|\nabla_\phi g_Q(z_t)\|^2}{2\mu_Q},$$

$$g_S(z_t) - g_S^*(x_t, \phi_t) \le \frac{\|\nabla_\psi g_S(z_t)\|^2}{2\mu_S}.$$

Here g_Q^* and g_S^* are the original global best-response values.

Corollary D.10 (Approximate Nash equilibrium under gradient dominance)**.** *Under the conditions of Theorem D.8, if the LL game satisfies Assumption D.9, let $\lambda > 0$, $\mu = \min\{\mu_Q, \mu_S\}$, and $\widehat{t} \in \arg\min_{0 \le t < T} \|\mathcal{H}_\lambda(z_t)\|$. Then $(\phi_{\widehat{t}}, \psi_{\widehat{t}})$ is an $\epsilon_{\mathrm{NE}}(T)$-Nash equilibrium of the LL game at $x_{\widehat{t}}$, where*

$$\epsilon_{\mathrm{NE}}(T) = \frac{1}{2\mu\lambda^2}\left(C_f + \frac{8\sqrt{2}L_H\|z_0 - z^*\|}{(1-\kappa L_H)^{3/2}\sqrt{T}}\right)^2. \tag{64}$$

Proof. By Assumption D.9 and the definition of $\mathcal{H}_\lambda$,

$$\begin{aligned}
&g_Q(z_t) - g_Q^*(x_t, \psi_t) + g_S(z_t) - g_S^*(x_t, \phi_t) \\
&\le \frac{1}{2\mu}\left(\|\nabla_\phi g_Q(z_t)\|^2 + \|\nabla_\psi g_S(z_t)\|^2\right) \\
&= \frac{1}{2\mu\lambda^2}\left\|[\mathcal{H}_\lambda(z_t)]_{(\phi,\psi)} - \nabla_{(\phi,\psi)} f(z_t)\right\|^2 \\
&\le \frac{\left(\|\mathcal{H}_\lambda(z_t)\| + C_f\right)^2}{2\mu\lambda^2},
\end{aligned}$$

where $\|\nabla_{(\phi,\psi)} f(z_t)\| \le C_f$ follows from the differentiability and C_f-Lipschitz continuity of f. Theorem D.8 gives

$$\|\mathcal{H}_\lambda(z_{\widehat{t}})\| \le \frac{8\sqrt{2}L_H\|z_0 - z^*\|}{(1-\kappa L_H)^{3/2}\sqrt{T}}.$$

Thus the sum of the two nonnegative best-response gaps at $z_{\widehat{t}}$ is at most $\epsilon_{\mathrm{NE}}(T)$. For every $\phi' \in \mathbb{R}^{d_\phi}$ and $\psi' \in \mathbb{R}^{d_\psi}$,

$$\begin{aligned}
g_Q(z_{\widehat{t}}) &\le g_Q^*(x_{\widehat{t}}, \psi_{\widehat{t}}) + \epsilon_{\mathrm{NE}}(T) \le g_Q(x_{\widehat{t}}, \phi', \psi_{\widehat{t}}) + \epsilon_{\mathrm{NE}}(T), \\
g_S(z_{\widehat{t}}) &\le g_S^*(x_{\widehat{t}}, \phi_{\widehat{t}}) + \epsilon_{\mathrm{NE}}(T) \le g_S(x_{\widehat{t}}, \phi_{\widehat{t}}, \psi') + \epsilon_{\mathrm{NE}}(T).
\end{aligned}$$

This proves the approximate Nash guarantee. □