TIGER: LEARNING FROM RESEARCHER TASTE TO GENERATE TAILORED RESEARCH IDEAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can generate research ideas, but their outputs often fail to match an individual researcher's taste. We introduce Tailored Idea GEnerator for Research (TIGER), a framework that learns individual preferences to guide idea generation. Using 400 real-world research requests and 2,000 candidate ideas, we collect ratings from three LLM researchers and two PhD researchers. For each researcher, we train a separate Qwen3.8-27B reward model (RM) on their annotations, then fix its parameters and use its scores to optimize a Qwen3-8B ideation policy through reinforcement learning. Neutral rewriting and auxiliary quality checks guide optimization alongside the preference reward. In pairwise comparisons where each judge chooses between an idea generated by its corresponding TIGER-8B policy and one generated by GPT-4.1, the three LLM researchers select the TIGER-8B idea in 56.6%-68.4% of 76 comparisons per researcher. PhD 1 and PhD 2 select the TIGER-8B idea in 19 of 30 pairs (63.3%) and 17 of 30 pairs (56.7%), respectively. All six directed cross-policy comparisons likewise favor the judge's matched TIGER-8B policy over an off-target policy, supporting the use of individual preference feedback for tailored research ideation.

1 INTRODUCTION

Recent work uses large language models (LLMs) to generate research ideas and hypotheses from scientific literature and observed data (Wang et al., 2024; Zhou et al., 2024; Liu et al., 2025; Baek et al., 2025; Lu et al., 2024; Yamada et al., 2025; Yu et al., 2024). Co-Scientist iteratively generates, critiques, and refines hypotheses (Gottweis et al., 2026), while NeuriCo connects literature review with experiment execution and analysis (Liu & Tan, 2026). These advances expand the research tasks that AI systems can support, but do not by themselves ensure that the proposed ideas reflect an individual researcher's priorities. Choosing which research directions to pursue involves *research taste*, a form of judgment discussed in human research and increasingly examined in LLMs (Olah, 2021; Mahajan et al., 2025; Tong et al., 2026).

Rather than imposing a single definition of a "good" research idea, our objective is to generate ideas that better match the target researcher's preferences, conditioned on a research request and literature context. The researcher may be a human researcher or an LLM; this formulation does not assume that their preferences coincide. We therefore ask: *Can a target researcher's research taste be learned from a fixed set of preference judgments and used to guide the generation of new research ideas?*

First, a major challenge is that collecting preference data for research ideation requires examples grounded in a concrete research context. Each example must pair a research request and its relevant literature with alternative ideas that the target researcher can meaningfully compare. Evaluating these alternatives requires understanding both the proposed contributions and their relation to prior work, making human annotation time-consuming. A practical approach must therefore learn effectively from a finite set of judgments that a researcher can reasonably provide.

To address this challenge, we propose TIGER (**T**ailored **I**dea **GE**nerator for **R**esearch), a framework that learns a researcher's taste from offline judgments and uses it to train a compact ideation policy. Figure 1 illustrates its two training stages. For research personalization, a key principle is that the judgments of a general-purpose LLM should not be assumed to reflect the preferences of a particular researcher (Dong et al., 2024; Chen et al., 2025b). TIGER therefore learns a researcher-specific reward model (RM) that captures these preferences and generalizes them to unseen query-idea pairs

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

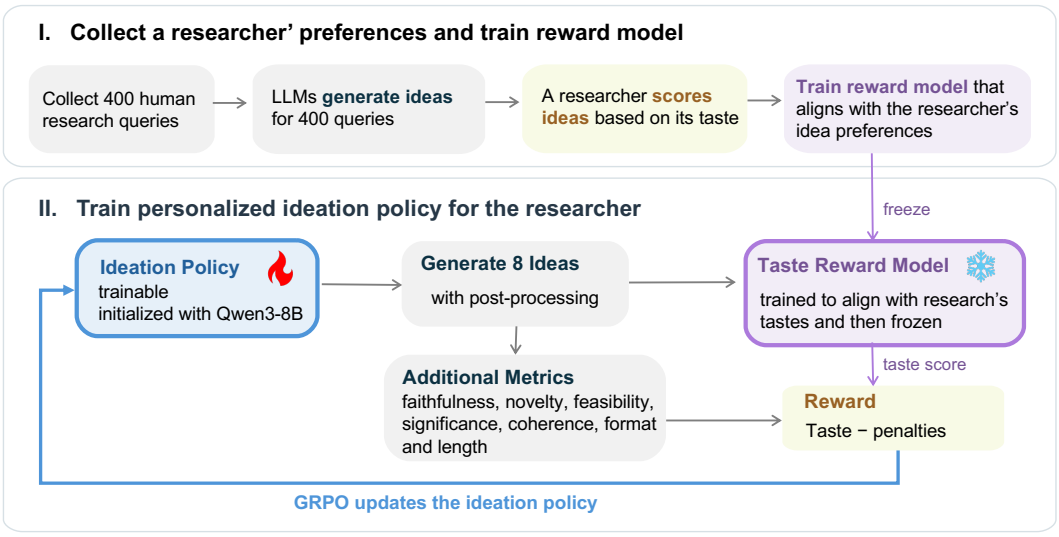

Figure 1: TIGER training pipeline, repeated for each researcher. (I) A researcher's ratings of candidate ideas generated from research requests train a reward model. (II) A policy initialized from Qwen3-8B samples eight ideas per request. Post-processing uses a frozen neutral rewriter to supply the RM input; a frozen quality judge scores five quality dimensions, with format and length checked separately. The taste score and penalties form the reward in Eq. 1, which drives GRPO updates on original-response tokens. Only the policy is updated; the RM, rewriter, and quality judge remain frozen.

for downstream policy training. Researchers may also prefer to keep unpublished ideas private rather than send them to an external API, further motivating a compact ideation model that can run locally while reflecting their research taste (e.g. an NVIDIA A100).

Therefore, first, we train a separate Qwen3.8-27B RM for each researcher using their annotations to predict both ratings and preference rankings. Second, we freeze the trained RM and use its scores as rewards to optimize a Qwen3-8B policy through reinforcement learning (RL). Thus, the RM learns the target researcher's taste through supervised training, while the policy learns to generate ideas that receive higher personalized rewards. In an initial study within our lab, two PhD students rated a pool of 227 candidate ideas spanning 40 research requests. Their agreement with LLM judges varied substantially across researchers (Appendix D, Figure 4), further motivating researcher-specific modeling rather than reliance on a single general-purpose judge.

Our contributions are as follows:

1) A benchmark for personalized research ideation, comprising 400 real-world research requests and 2,000 candidate ideas generated by five LLM backbones. The benchmark includes preference annotations from three LLM researchers and two PhD researchers, each providing ratings over 2,000 ideas, enabling systematic study of researcher-specific taste.

2) A researcher-specific reward-modeling and policy-optimization framework that converts a finite set of offline preference judgments into a compact personalized ideation model. For each researcher, we train a Qwen3.8-27B reward model on within-request rankings and numerical ratings, then freeze it to optimize a Qwen3-8B policy model with predicted rewards, neutral rewriting, and auxiliary quality penalties, requiring no additional researcher annotation during policy training.

3) An evaluation protocol for tailored research ideation that combines comparisons against GPT-4.1 with cross-policy judgments to test both overall quality and researcher-specific alignment. Under this evaluation, the corresponding TIGER-8B policies achieve win-rate point estimates above 50% for all three LLM researchers, while both PhD researchers prefer their corresponding policies in a majority of direct comparisons with GPT-4.1. Cross-policy judgments further show that each LLM researcher favors its matched policy over policies trained for the other researchers.

2 Related Work

2.1 Autonomous Researcher

Automatic idea generation is becoming an important first step toward AI Researchers and has attracted increasing attention. Existing work grounds hypothesis generation in research literature or observed data (Wang et al., 2024; Liu et al., 2025; Zhou et al., 2024). More comprehensive systems, such as AI Co-Scientist (Gottweis et al., 2026) and NeuriCo (Liu & Tan, 2026), extend this process to multiple stages of research, from literature review to experimental design. To improve research quality, GIANTS (He-Yueya et al., 2026) predicts core insights by combining two parent papers, while EvoIdeator (Sauter et al., 2026) uses checklist-grounded feedback in a reinforcement learning pipeline. However, existing work primarily focuses on generating research ideas, while paying limited attention to judging whether the generated ideas align with a particular researcher's preferences.

2.2 AI Research Taste

Research taste has been studied through citation-based measures (Koltun & Hafner, 2021) and institutional publication records (Gong et al., 2026). AI Can Learn Scientific Taste (Tong et al., 2026) learns a scientific judge from citation-based community preferences and uses it as a reward model to train an idea generator with reinforcement learning. TIGER instead learns a separate reward model from each researcher's own ratings of candidate ideas and uses it to align the corresponding ideation policy with that researcher's preferences. Personalized Auto-Research (Ni et al., 2026) further proposes conditioning research workflows on a researcher's prior work, methods, and collaborators. LLM-generated ideas also differ systematically from human research ideas in how they frame opportunities and construct contributions (Chen et al., 2026b). Benchmarks evaluate both hypothesis generation (Liu et al., 2026) and the alignment of idea evaluations with expert judgments (Wang et al., 2026; Chen et al., 2026a). However, human studies and expert evaluations still identify limitations in LLM self-evaluation and agreement with expert preferences (Si et al., 2025; Chen et al., 2026a).

2.3 Preference Alignment

Preference learning uses human or AI feedback to guide model behavior (Christiano et al., 2017; Lee et al., 2024), with personalized approaches further accounting for differences between individual users (Li et al., 2024; Chen et al., 2025a; Barreto et al., 2025). These methods provide a basis for learning research taste. Our setting requires learning a particular researcher's preferences over research ideas from a fixed set of judgments and using them to guide new generations, without collecting further judgments from that researcher during policy training.

Reward design also depends on the task. Mathematical reasoning and code generation can use correctness checks (Lightman et al., 2024; Le et al., 2022), while agent reward models assess progress toward a task goal (Xi et al., 2026). Research ideas require judgments about their proposed contributions and relation to existing literature, without a single correctness criterion that captures a researcher's preferences. Preference models and task-specific rubrics for open-ended generation (Jia et al., 2025; Shen et al., 2026) provide ways to learn such reward signals.

Finally, optimizing an imperfect reward model can increase predicted scores without improving the outputs according to the target researcher (Gao et al., 2023; Eisenstein et al., 2024). In research ideation, a policy may receive high rewards for polished descriptions, repackaged literature, or unsupported claims of completed experiments. We therefore combine preference rewards with neutral rewriting and auxiliary quality checks.

3 Tailored Idea GEnerator for Research (TIGER)

TIGER first learns a reward model from a researcher's annotations and then uses the trained model to optimize an ideation policy (Figure 1). We describe data collection, supervised RM training, and policy optimization in that order.

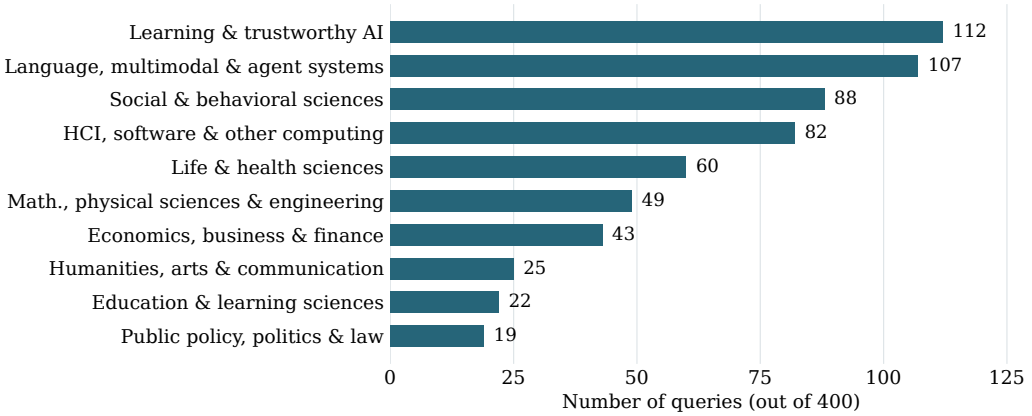

Figure 2: Research areas covered by the 400 requests, using LLM-assisted labels. Queries may belong to multiple areas. HCI denotes human–computer interaction.

3.1 Data Collection

We collect 400 real-world research requests submitted to Hypogenic AI. We use the same ideation agent with five LLM backbones: Qwen3-8B, DeepSeek V4.1 Flash, Qwen3-32B, Llama 3.3-70B Instruct, and GPT-5.6 Sol. The agent generates ideas from each request and its literature review. We retain 2,000 ideas, with 400 contributed by each backbone and five candidates per request. Before evaluation, we group candidates by request and hide their model identities and generation order.

The requests span learning and trustworthy AI; language, multimodal, and agent systems; social and behavioral sciences; HCI, software, and other computing; life and health sciences; mathematics, physical sciences, and engineering; economics, business, and finance; humanities, arts, and communication; education and learning sciences; and public policy, politics, and law. Figure 2 shows the coverage of these ten categories; a request may belong to multiple categories.

The three LLM researchers and two PhD researchers each rate 2,000 ideas for the same 400 requests, and their reward models use these annotations. In the human interface, researchers see a research query, its literature context, and several candidate ideas. They rate each idea on a five-point preference scale from [-2, -1, 0, 1, 2], indicating how much they like or dislike it. These ratings capture each researcher's overall preference for an idea. We retain each researcher's ratings separately to study individual preferences. These full-pool annotations are distinct from the pilot ratings in Figure 4.

LLM researchers follow a shared evaluation protocol with the same scoring dimensions, score scales, and output format. We retain each researcher's component scores and sum them to obtain its overall score for an idea. The researchers, scoring rubric, and prompts used are described in Section 4.1.

We operationalize research taste through a researcher's relative preferences among candidate ideas proposed for the same query. For human researchers, these preferences are obtained from overall ratings; for LLM researchers, they are obtained from summed component scores. Using the same candidate pool and evaluation protocol across LLMs helps separate differences in their preferences from differences in task presentation. The exploratory study in Figure 4(a) shows that agreement varies across LLM pairs, with within-query Spearman correlations ranging from 0.51 to 0.62.

3.2 Learning Researcher-Specific Reward Models

We initialize a separate reward model from Qwen3.8-27B for each researcher and train it on that researcher's offline annotations. The model learns to predict the researcher's numerical ratings and preference rankings, yielding a scalar taste reward for a request–idea pair. For example, the RM trained on Haiku's ratings is a Qwen3.8-27B model supervised by annotations from Haiku. Researcher names identify the source of preference supervision; all five RMs share the Qwen3.8-27B backbone. We call a Qwen3-8B policy optimized with this RM the *Haiku-aligned policy*, and use the same convention for the other researchers.

Algorithm 1 Training a Researcher-Specific Reward Model

Require: Labels $\mathcal{D}_e = \{(q_i, \mathcal{I}_i, \mathbf{S}_i^e)\}_{i=1}^N$; D-output reward model f_{θ_e}
Require: Score scaling g_e; temperature $T > 0$; weights $\lambda_{\text{total}}, \lambda_{\text{dim}} \geq 0$
for each query batch $B \subset \mathcal{D}_e$ **do**
  for each query i in B **do**
    $\mathbf{y}_{i,j}^e \leftarrow g_e(\mathbf{s}_{i,j}^e)$, $\hat{\mathbf{y}}_{i,j} \leftarrow f_{\theta_e}(q_i, I_{i,j})$ for all j
    $y_{i,j}^e \leftarrow \sum_{d=1}^D y_{i,j,d}^e$, $r_{i,j} \leftarrow \sum_{d=1}^D \hat{y}_{i,j,d}$ for all j
    $p_i^e \leftarrow \text{softmax}\left((y_{i,1}^e, \ldots, y_{i,K_i}^e)/T\right)$
    $\hat{p}_i \leftarrow \text{softmax}\left((r_{i,1}, \ldots, r_{i,K_i})/T\right)$
    $\mathcal{L}_{\text{rank}}^{(i)} \leftarrow -\sum_{j=1}^{K_i} p_{i,j}^e \log \hat{p}_{i,j}$
    $\mathcal{L}_{\text{total}}^{(i)} \leftarrow \frac{1}{K_i} \sum_{j=1}^{K_i} \ell_{\text{Huber}}(r_{i,j}, y_{i,j}^e)$
    $\mathcal{L}_{\text{dim}}^{(i)} \leftarrow \frac{1}{DK_i} \sum_{j=1}^{K_i} \sum_{d=1}^{D} \ell_{\text{Huber}}(\hat{y}_{i,j,d}, y_{i,j,d}^e)$
    $\mathcal{L}_i \leftarrow \mathcal{L}_{\text{rank}}^{(i)} + \lambda_{\text{total}} \mathcal{L}_{\text{total}}^{(i)} + \lambda_{\text{dim}} \mathcal{L}_{\text{dim}}^{(i)}$
  end for
  Update θ_e to minimize $|B|^{-1} \sum_{i \in B} \mathcal{L}_i$
end for
return f_{θ_e}

Supervision. For each query q_i, researcher e scores a candidate set $\mathcal{I}_i = \{I_{i,1}, \ldots, I_{i,K_i}\}$. Here, q_i includes the literature context when it is supplied to the reward model. The annotation $\mathbf{s}_{i,j}^e$ contains one overall rating or D component scores for idea $I_{i,j}$. We keep each researcher's labels separate and apply a scaling factor g_e for scores in training. The model f_{θ_e} predicts the corresponding scores and sums them to a scalar reward. An overall rating uses a prediction head, corresponding to $D = 1$. Here, N denotes the number of training queries and B denotes a minibatch of queries.

Training objective. A listwise soft-label cross-entropy loss learns relative preferences among ideas for the same query. Equal total ratings receive equal target probabilities. A total-score Huber loss fits the researcher's numerical scores. When component labels are available, we additionally fit them with a component-level Huber loss. For overall ratings, this component loss is omitted by setting $\lambda_{\text{dim}} = 0$. Algorithm 1 gives the shared procedure; the score scaling, prediction head, temperature, and loss weights follow the annotation format and experimental setting.

Each query's candidates are kept together during training, and data splits separate entire queries. After supervised training and checkpoint selection, we freeze the RM: its learned parameters remain fixed while it scores new ideas during policy optimization. Personalization is therefore learned before freezing, and subsequent RL updates only the ideation policy. Section 4.1 describes the experimental setup, and Appendix C provides the model heads, label encodings, and training configurations.

3.3 Optimizing the Ideation Policy

For each target researcher, we initialize a separate ideation policy from Qwen3-8B and optimize it using the RM trained on that researcher's annotations. Given a research request q_i and its literature review L_i, the policy samples a group of candidate ideas $\{I_{i,j}\}_{j=1}^K$. Each candidate receives a scalar reward, and the policy is optimized from rewards relative to the other candidates generated for the same request. The trained RM supplies taste feedback with fixed parameters, without further judgments from the researcher.

Rollout feedback. A frozen rewriting model converts each candidate $I_{i,j}$ into a neutrally worded version $\widetilde{I}_{i,j}$ while preserving its scientific content. The researcher-specific reward model assigns a taste score to the rewritten idea, using the same input context as during RM training. Separately, a frozen quality judge evaluates the candidate with (q_i, L_i) for faithfulness, novelty, feasibility, significance, and coherence. We also apply penalties for invalid output format and excessive length. The resulting reward is

$$R = s_{\text{taste}} - p_{\text{faith}} - p_{\text{nov}} - p_{\text{feas}} - p_{\text{sig}} - p_{\text{coh}} - p_{\text{fmt}} - p_{\text{len}}. \tag{1}$$

Reward components. The taste reward s_{taste} measures how well an idea matches the target researcher's preferences. It is computed from the neutrally rewritten idea so that persuasive wording alone does not increase the reward.

The quality penalties target specific weaknesses in a proposed research idea. The faithfulness penalty p_{faith} applies when the response invents unsupported research gaps, misrepresents prior work, or states expected outcomes as established results. The novelty penalty p_{nov} applies when the idea largely repeats or repackages existing work. The feasibility penalty p_{feas} applies when the proposed study is difficult to execute or evaluate. The significance penalty p_{sig} applies when the idea would offer limited value even if successful. The coherence penalty p_{coh} applies when the research question, claimed contribution, methodology, and evaluation plan are not well aligned.

The format penalty p_{fmt} is applied when the response does not follow the required structure, while the length penalty p_{len} discourages the policy from obtaining a higher reward simply by generating a longer response. Format and length penalties are computed from the sampled response. Section 4.2 gives the exact penalty values and training settings.

Training objective. We optimize the policy with group relative policy optimization (GRPO). For each request, rewards are normalized within its candidate group to compute relative advantages. These advantages weight a PPO-style clipped objective over the tokens of the original sampled responses. Groups with identical rewards do not contribute to the update. The reward model, rewriting model, and quality judge remain fixed throughout policy training.

We also consider variants without the taste reward, without the frozen quality judge, and without neutral rewriting. These ablations test the roles of researcher-specific taste, auxiliary judging, and presentation style, respectively; Section 4.3 gives the full settings and results.

4 Evaluation

Our evaluation asks whether TIGER can learn a researcher's preferences and use them to generate ideas that the researcher prefers. We compare matched and off-target reward models on validation ratings and evaluate generated ideas against GPT-4.1. Our cross-policy evaluation further examines whether each LLM researcher prefers their matched policy. Ablations examine the roles of the taste reward, neutral rewriting, and auxiliary judging.

4.1 Reward Model

Researchers and annotation prompts. The human setting uses individual preference ratings from two PhD researchers, denoted PhD 1 and PhD 2, each on 2,000 ideas as described in Section 3.1. The three LLM researchers are GPT-5.6 Terra, Gemini 3.7 Flash, and Claude Haiku 4.5. Each LLM scores novelty, significance, plausibility, feasibility, expected effectiveness, and excitement. The prompts request integer scores from 0 to 100, permit ties, and ask researchers to assess the contribution rather than persuasive wording. Novelty is scored both closed-book and with literature, while expected effectiveness uses literature; the remaining dimensions are scored closed-book. This yields seven scores per idea, with a total ranging from 0 to 700.

Experimental setup. We train a separate Qwen3.8-27B reward model for each researcher using 320 training queries and 80 validation queries. RMs trained on human ratings receive the request and idea, while those trained on LLM ratings also receive the literature review. Appendix C provides the model heads, label encodings, optimization settings, checkpoint selection, and score calibration.

Comparison and metrics. We report pooled strict pair accuracy: the number of correctly ordered idea pairs divided by the number of eligible pairs across queries. An eligible pair has the same strict order in every reference ordering; a predicted tie counts as incorrect. The human labels treat identical ratings as ties. For LLM labels, we sort scores in descending order and form tie groups within five points of each group's highest score on the original 0–700 scale.

To test whether a reward model learns individual preferences, we compare the matched RM, trained on the target researcher's ratings, with off-target RMs trained on other researchers' ratings. For each target researcher, all RMs score the same validation candidates and are evaluated against that researcher's ordering. Table 1 reports each comparison separately. We estimate matched-minus-off-target differences with 10,000 paired bootstrap resamples of queries. All models use the same 80 validation queries. Each RM retains its native input format: those trained on LLM ratings receive the literature review, while those trained on human ratings receive only the request and idea. Comparisons

within each group share the input format and training candidate pool; comparisons across groups also reflect these differences.

Results. The matched RM achieves the highest strict pair accuracy for each of the five target researchers. On Haiku's labels, the RM trained on Haiku's ratings outperforms the RM trained on Terra's ratings by 9.03 percentage points (95% CI: 5.58-12.52). On Gemini's labels, the matched RM outperforms the RM trained on Terra by 8.35 percentage points (95% CI: 5.63-11.15).

4.2 Policy Training and Evaluation

Training settings. For each target researcher, we initialize the policy from Qwen3-8B with thinking disabled and train a separate LoRA adapter on all linear modules using the matched reward model. Each rollout generates one idea as a JSON array containing its title, description, and summary (`tldr`), with a target length of at most 2,000 characters. Optimization settings, including the configuration-specific LoRA scaling, are listed in Table 4 in Appendix A.

Table 1: Strict pair accuracy (%) of matched and off-target RMs, all initialized from Qwen3.8-27B. Rows identify the researcher whose labels are used for evaluation; columns identify whose annotations trained the RM. Bold diagonal entries are matched RMs.

	Source of RM training annotations				
Target researcher	Terra	Gemini 3.7	Haiku	PhD 1	PhD 2
Terra	**77.35**	76.01	76.81	67.29	59.92
Gemini 3.7	83.55	**91.90**	91.26	71.08	60.28
Haiku	82.71	90.71	**91.74**	71.74	60.00
PhD 1	67.11	60.49	54.44	**76.56**	74.67
PhD 2	61.96	52.97	45.81	74.64	**79.96**

Data and auxiliary models. For policy training, we split the 400 requests into 324 training queries and 76 held-out test queries. Policy optimization uses only the 324 training queries; the remaining 76 queries are reserved for the final pairwise evaluation reported below. Across all policy-training runs, both the neutral rewriter and quality judge are frozen Qwen3.8-27B models. The policy prompt requests a concrete literature-grounded contribution, its distinction from the closest work, and a credible evaluation plan, while keeping expected outcomes hypothetical. The rewriter removes promotional wording and unsupported certainty while preserving scientific content, and the quality judge scores faithfulness, novelty, feasibility, significance, and coherence from 0 to 4 in a single evaluation.

Penalty settings. In this auxiliary-quality configuration, a frozen quality judge scores faithfulness, novelty, feasibility, significance, and coherence. It returns an integer score $s_d \in \{0, 1, 2, 3, 4\}$ for each dimension. We convert it to a penalty $p_d = (4 - s_d)/4$, so each dimension contributes between zero and one. An invalid output format incurs a penalty of 2. The length penalty is zero up to 2,000 characters, increases linearly to 2 at 4,000 characters, and remains capped at 2. Format and length checks use the sampled response. The taste score uses the rewritten idea and the frozen reward model, with score calibration described in Appendix C.

Preference scores. For each of the three LLM-aligned policies, we compare the mean taste score of ideas sampled in the initial training batch with that of ideas generated by the corresponding trained TIGER-8B policy, using the same researcher-specific RM for both. The mean score increases from 4.00 to 4.57 for Terra, from 3.64 to 4.77 for Haiku, and from 3.75 to 4.85 for Gemini, corresponding to increases of 0.57, 1.13, and 1.10, respectively. Each RM is held fixed throughout the corresponding policy's training and scoring. All RM scores are measured before subtracting quality, format, and length penalties.

Qualitative example. Figure 3 reproduces the titles and summaries of ideas generated before and after RL on the same request. The later idea specifies a comparison between explicit and implicit state-transition rules, with robustness under perturbations as an evaluation target. Appendix B provides the complete original descriptions.

Independent evaluation. We evaluate generated ideas independently of the learned reward scores through pairwise comparisons with GPT-4.1. Each comparison pairs ideas generated for the same research request and literature review, with model identities hidden. The target researcher judges the paired ideas using the question "which idea do you prefer?" We report the number of evaluated comparisons and the policy's win rate.

Request: *generate ideas on synthesizing LLM tool learning environments*

Qwen3-8B (before RL) **RM score: 3.27**	**Haiku-aligned TIGER-8B** **RM score: 4.64**
Synthesizing Tool-Use Environments with Dynamic State and Multi-Agent Interaction	*Evaluating the Role of State Transition Fidelity in Tool-Use Environment Synthesis*
Develop dynamic, multi-agent tool-use environments to enhance agent training through evolving state and interaction complexity.	Compare agent performance on tool-use tasks using environments with explicit vs. implicit state transition rules, testing whether fidelity to programmatic state validity reduces adversarial/synthetic perturbation vulnerability.

Figure 3: Verbatim titles and summaries from model outputs for the same request, literature review, and generation prompt. The RM trained on Haiku's ratings scores neutral rewrites of the complete ideas at 3.27 and 4.64, before penalties. Appendix B reproduces the full descriptions.

Generation results. Each LLM researcher makes 76 pairwise judgments of its corresponding TIGER-8B policy against GPT-4.1. Terra prefers TIGER in 46 pairs (60.5%), Gemini in 52 pairs (68.4%), and Haiku in 43 pairs (56.6%). In direct human evaluation, PhD 1 prefers their corresponding TIGER-8B policy in 19 of 30 pairs (63.3%) and GPT-4.1 in the remaining 11. PhD 2 prefers their corresponding policy in 17 of 30 pairs (56.7%) and GPT-4.1 in the remaining 13. Table 2(a) reports the completed LLM and human comparisons.

We additionally compare the Terra-aligned TIGER-8B policy with two base Qwen3-8B baselines on the same 76 held-out requests. The first uses direct generation; the second samples eight ideas and selects the one with the highest score from the RM trained on Terra's ratings (best-of-8). Against each baseline, Terra prefers TIGER-8B in 60 pairs (78.9%) and the baseline in 16 pairs (21.1%). Table 2(c) summarizes these comparisons.

Cross-policy evaluation. We compare policies aligned with Terra, Gemini 3.7, and Haiku on 30 shared research requests with their literature reviews. For each policy pair, the corresponding LLM researchers judge neutral rewrites of the paired ideas using the question "which idea do you prefer?" Each of the six directed comparisons contains one judgment per request, yielding 30 judgments per direction and 180 judgments in total, with no ties. Table 2(b) reports the off-target policy win rate: each cell measures how often the row judge prefers the column policy over its own matched policy. Lower values indicate stronger preference for the researcher-matched policy.

Under Terra judgments, the Gemini- and Haiku-aligned policies win 11 of 30 pairs (36.7%) and 8 of 30 pairs (26.7%) against the Terra-aligned policy, respectively. Across all six directed comparisons, off-target policy win rates range from 3.3% to 36.7%, remaining below 50% in every case. These results support researcher-specific policy preferences.

4.3 Ablation Study

Ablation settings. We consider three variants of the Terra-aligned policy that each remove one source of feedback from the complete pipeline. *Without the reward model*, the policy is trained only with the five quality penalties and the format and length penalties, isolating the contribution of researcher-specific taste. *Without the quality judge*, the policy retains the taste reward together with the format and length penalties but removes the five quality penalties, testing whether auxiliary judging constrains shortcut behaviors. *Without neutral rewriting*, the complete reward is retained, but the RM scores the original policy response, testing sensitivity to presentation style. Table 3(a) summarizes the components used by each variant.

Terra ablation results. We evaluate the contribution of individual TIGER components by comparing full TIGER with three ablated variants on the 76 shared test requests. In each pairwise comparison, Terra is provided with the complete request and literature review, along with two anonymous ideas (Idea A/B), and asked to select the preferred idea. The order of presentation is randomized to avoid positional bias. Full TIGER substantially outperforms the variant without the quality judge, winning 69/76 comparisons (90.8%). It also wins 51/76 comparisons (67.1%) against the variant without the reward model and 45/76 comparisons (59.2%) against the variant without rewriting.

Table 2: Policy preference evaluation. (a) Pairwise preferences for TIGER-8B versus GPT-4.1. Both PhD researchers provide direct human judgments. (b) Off-target policy win rate (%, ↓), based on 30 judgments per direction on shared requests. Each cell is the column policy's win rate against the row judge's matched policy; lower values indicate stronger preference for the matched policy. Opposite directions use different judges, and dashes indicate self-comparisons. (c) Terra preferences for TIGER-8B versus direct generation and personalized-RM best-of-8 selection from base Qwen3-8B.

(a) Comparison with GPT-4.1

Judge	TIGER wins	GPT wins	TIGER (%)
Terra	46	30	60.5
Gemini 3.7	52	24	68.4
Haiku	43	33	56.6
PhD 1	19	11	63.3
PhD 2	17	13	56.7

(b) Off-target win rate (%, ↓)

	Off-target policy trained for		
Judge	Terra	Gemini 3.7	Haiku
Terra	—	36.7	26.7
Gemini 3.7	3.3	—	36.7
Haiku	33.3	36.7	—

(c) Base Qwen3-8B baselines under Terra judgments

Baseline	TIGER wins	Baseline wins	TIGER (%)
Direct generation	60	16	78.9
Personalized-RM best-of-8	60	16	78.9

Panel (a): Each LLM judge evaluates 76 idea pairs; each PhD researcher evaluates 30 pairs. Panel (b): one judgment per request in each of six directions, giving 30 judgments per direction and 180 in total; no ties. Panel (c): Terra evaluates 76 pairs per baseline; best-of-8 selects with the RM trained on Terra's ratings.

Table 3: RL ablations for Terra. (a) Training components; all variants retain format and length penalties. (b) Pairwise Terra preferences between each ablation and full TIGER. Each row contains 76 judgments; full preference is full TIGER wins divided by 76.

(a) Ablation design

Variant	Taste reward	Quality judge	Rewrite before RM
Full TIGER	Yes	Yes	Yes
Without RM	No	Yes	N/A
Without quality judge	Yes	No	Yes
Without rewriting	Yes	Yes	No

(b) Comparison with full TIGER

Ablated variant	Ablation wins	Full wins	Full preference (%)
Without quality judge	7	69	90.8
Without rewriting	31	45	59.2
Without RM	25	51	67.1

5 Conclusion and Limitations

TIGER adapts a compact 8B model to generate research ideas tailored to an individual researcher's taste. TIGER learns a separate reward model from each researcher's offline judgments to guide policy optimization. The matched reward model achieves the highest strict pair accuracy for each of the five researchers. Neutral rewriting and auxiliary quality checks guide generation alongside the learned preference reward. In comparisons with GPT-4.1, the corresponding TIGER-8B policies win 46/76 Terra judgments, 52/76 Gemini judgments, and 43/76 Haiku judgments. Terra also prefers its TIGER-8B policy over both direct generation and personalized-RM best-of-8 selection from the base Qwen3-8B model in 60/76 comparisons. In the Terra ablations, full TIGER wins 69/76 comparisons against the variant without the quality judge, 51/76 against the variant without the reward model, and 45/76 against the variant without rewriting. Direct human evaluation favors the corresponding policies in 19 of 30 pairs (63.3%) for PhD 1 and 17 of 30 pairs (56.7%) for PhD 2. Cross-policy comparisons further support researcher-specific generation: off-target policies win 3.3%–36.7% of comparisons against the judge's matched policy. These results support learning individual research preferences as a way to tailor idea generation to the researcher who will assess and pursue the proposed work.

Our evaluation focuses on preference alignment across five researchers and 400 research requests. Extending TIGER to broader research communities and evolving researcher preferences is an important direction for future work. Assessing the scientific impact of generated ideas requires carrying out the proposed studies. Further evaluation should also examine content preservation during rewriting and the reliability of auxiliary quality judgments.

AI USE STATEMENT

We employ large language models (LLMs) primarily to improve the grammar and clarity of our writing; the first draft of the paper was written entirely by humans. All research ideas, directions, and decisions, however, are independently conceived and carried out by the authors. We use AI tools (Codex and Claude Code) for vibe coding under the strict guidance of the authors, and we manually double-check and verify the content generated by AI.

ETHICS STATEMENT

TIGER aims to support personalized research ideation, but its ability to learn individual preferences could also be exploited to align idea generation with malicious objectives. The framework could therefore be misused to develop and disseminate ideas intended to cause harm. Our quality checks assess scientific quality and do not guarantee benign intent or use; human oversight remains necessary to assess potential harms before pursuing or disseminating generated ideas.

REPRODUCIBILITY STATEMENT

Section 3.1 describes data collection, and Section 4.1 specifies the reward-model evaluation setup and metrics. Appendix C provides the reward-model architecture, label encodings, optimization settings, checkpoint selection, and score calibration. Section 4.2 details policy-training inputs, auxiliary models, reward penalties, and evaluation procedures. Appendix A lists the policy optimization settings. Appendix F reproduces the generation, rewriting, quality-judging, and preference-evaluation prompts, together with the learned RM input formats.

REFERENCES

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. Researchagent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 conference of the nations of the Americas chapter of the association for computational linguistics: human language technologies (volume 1: long papers)*, pp. 6709–6738, 2025.

André Barreto, Vincent Dumoulin, Yiran Mao, Mark Rowland, Nicolas Perez-Nieves, Bobak Shahriari, Yann Dauphin, Doina Precup, and Hugo Larochelle. Capturing individual human preferences with reward features. In *Advances in Neural Information Processing Systems*, volume 38, pp. 82119–82154. Curran Associates, Inc., 2025. doi: 10.52202/085713-2749. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/761bc7f095f927e5a5de31ed4154bb45-Abstract-Conference.html.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=1kFDrYCuSu.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Vinayak. Pal: Sample-efficient personalized reward modeling for pluralistic alignment. In *International Conference on Learning Representations*, volume 2025, pp. 15755–15799, 2025b.

Zhiyu Chen, Keyu Zhao, Jigao Fu, Dong Liang, Yanbiao Wu, Jiaoyang Li, Haidong Xue, Xinhua Zeng, Yuanyi Zhen, Fengli Xu, and Yong Li. Ideation Arena: Evaluating LLM Generated Research Ideas with Battle-style Human Expert Assessment, 2026a. URL https://arxiv.org/abs/2608.29696. arXiv:2608.29696.

Ziyu Chen, Yilun Zhao, and Arman Cohan. Measuring the Gap Between Human and LLM Research Ideas, 2026b. URL https://arxiv.org/abs/2607.01233. arXiv:2607.01233.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL

https://proceedings.neurips.cc/paper_files/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Yijiang River Dong, Tiancheng Hu, and Nigel Collier. Can llm be a personalized judge? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10126–10141, 2024.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5u1GpUkKtG.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Ziqin Gong, Ning Li, and Huaikang Zhou. LLMs learn scientific taste from institutional traces across the social sciences, 2026. URL https://arxiv.org/abs/2603.16659. arXiv:2603.16659.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomašev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, José R. Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *Nature*, 655:487–496, 2026. doi: 10.1038/s41586-026-10644-y. URL https://www.nature.com/articles/s41586-026-10644-y.

Joy He-Yueya, Anikait Singh, Ge Gao, Michael Y. Li, Sherry Yang, Chelsea Finn, Emma Brunskill, and Noah D. Goodman. GIANTS: Generative insight anticipation from scientific literature. In *Second Workshop on Language Models for Scientific Discovery (LM4Sci), COLM*, 2026. doi: 10.48550/arXiv.2604.09793. URL https://arxiv.org/abs/2604.09793. To appear.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-Zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *arXiv preprint arXiv:2506.00103*, 2025. doi: 10.48550/arXiv.2506.00103. URL https://arxiv.org/abs/2506.00103.

Vladlen Koltun and David Hafner. A Measure of Research Taste, 2021. URL https://arxiv.org/abs/2105.08089. arXiv:2105.08089.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 21314–21328. Curran Associates, Inc., 2022. doi: 10.52202/068431-1549. URL https://papers.nips.cc/paper_files/paper/2022/hash/8636419dea1aa9fbd25fc4248e702da4-Abstract-Conference.html.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26874–26901. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24t.html.

Xinyu Li, Ruiyang Zhou, Zachary C. Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. In *NeurIPS 2024 Workshop on Adaptive Foundation Models*, 2024. URL https://openreview.net/forum?id=xxBoca28oG.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Haokun Liu and Chenhao Tan. NeuriCo: Towards reliable AI scientists. In *ICML 2026 AI for Science Workshop*, 2026. URL https://openreview.net/forum?id=yr0OIxGNWI.

Haokun Liu, Yangqiaoyu Zhou, Mingxuan Li, Chenfei Yuan, and Chenhao Tan. Literature meets data: A synergistic approach to hypothesis generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 245–281, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.12. URL https://aclanthology.org/2025.acl-long.12/.

Haokun Liu, Sicong Huang, Jingyu Hu, Yangqiaoyu Zhou, and Chenhao Tan. HypoBench: Towards Systematic and Principled Benchmarking for Hypothesis Generation, 2026. URL https://arxiv.org/abs/2504.11524. arXiv:2504.11524.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

Parv Mahajan, Yilin, and yix. TastyBench: Toward measuring research taste in LLM. https://www.lesswrong.com/posts/Mxsy7wYvsCRv5dGrw/tastybench-toward-measuring-research-taste-in-llm, December 2025. LessWrong research update, December 2, 2025.

Bo Ni, Franck Dernoncourt, Hongjie Chen, Yu Wang, Nesreen K. Ahmed, Zhengzhong Tu, Tyler Derr, and Ryan A. Rossi. Personalized Auto-Research: Towards a True AI Co-Scientist, 2026. URL https://arxiv.org/abs/2608.14881. arXiv:2608.14881.

Chris Olah. Research taste exercises. https://colah.github.io/notes/taste/, January 2021. Blog post, January 9, 2021.

Andreas Sauter, Yuyue Zhao, Jacopo Urbani, Wenxiang Hu, Zaiqiao Meng, Lun Zhou, Xiaohui Yan, and Yougang Lyu. EvoIdeator: Evolving scientific ideas through checklist-grounded reinforcement learning. *arXiv preprint arXiv:2603.21728*, 2026. doi: 10.48550/arXiv.2603.21728. URL https://arxiv.org/abs/2603.21728.

William F. Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks. *arXiv preprint arXiv:2602.05125*, 2026. doi: 10.48550/arXiv.2602.05125. URL https://arxiv.org/abs/2602.05125.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=M23dTGWCZy.

Jingqi Tong, Mingzhe Li, Hangcheng Li, Yongzhuo Yang, Yurong Mou, Weijie Ma, Hongji Chen, Xiaoran Liu, Qinyuan Cheng, Ming Zhang, Qiguang Chen, Weifeng Ge, Qipeng Guo, Tianlei Ying, Tianxiang Sun, Yining Zheng, Zhiheng Xi, Xinchi Chen, Jun Zhao, Ning Ding, Xuanjing Huang, Yu-Gang Jiang, and Xipeng Qiu. AI Can Learn Scientific Taste, 2026. URL https://arxiv.org/abs/2603.14473. arXiv:2603.14473.

Chenrun Wang, Mingxuan Zhu, Tiancheng Huang, Wenjie Li, Yujie Zhang, Zichen Zhu, Zhiying Zou, Kai Yu, and Lu Chen. LigBench: A Unified and Human-Aligned Benchmark for LLM-based Research Idea Generation, 2026. URL https://arxiv.org/abs/2608.13136. arXiv:2608.13136.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 279–299, Bangkok, Thailand, August

2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.18. URL https://aclanthology.org/2024.acl-long.18/.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentPRM: Process reward models for LLM agents via step-wise promise and progress. In *Proceedings of the ACM Web Conference 2026*, pp. 4184–4195. Association for Computing Machinery, 2026. doi: 10.1145/3774904.3792551. URL https://doi.org/10.1145/3774904.3792551.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Haofei Yu, Zhaochen Hong, Zirui Cheng, Kunlun Zhu, Keyang Xuan, Jinwei Yao, Tao Feng, and Jiaxuan You. Researchtown: Simulator of human research community. *arXiv preprint arXiv:2412.17767*, 2024.

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 117–139, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4science-1.10. URL https://aclanthology.org/2024.nlp4science-1.10/.

A Policy Optimization Settings

Table 4 summarizes the optimization settings for all five researcher-aligned policies, as described in Section 4.2. The policies aligned with PhD 1 and PhD 2 use the same optimization hyperparameters as those aligned with Terra and Haiku, including LoRA rank 32 and alpha 64. The Gemini-aligned policy uses LoRA alpha 32. The learning rate remains constant after warmup. Appendix F lists the shared V7 policy prompt and the auxiliary-model instructions.

Table 4: Policy optimization settings for all five researchers.

Setting	Value
Policy backbone	Qwen3-8B
Rollouts per request	8
Rollout temperature	1.0
Request groups per update	48
Optimization epochs per update	1
LoRA rank	32
LoRA alpha (Terra / Haiku)	64
LoRA alpha (PhD 1 / PhD 2)	64
LoRA alpha (Gemini)	32
Learning rate	3×10^{-5}
Warmup updates	10
PPO clipping range	$[0.8, 1.28]$
Rollout importance-weight cap	2.0
Maximum prompt / response tokens	16,384 / 2,048
KL coefficient / entropy bonus	0 / 0

Reward composition across researchers. All five policies use their corresponding RM score with coefficient one and subtract five auxiliary quality penalties, each with coefficient one. Each quality penalty is $(4 - s_d)/4 \in [0, 1]$; the format and length penalties each have a maximum value of 2. We use the same penalty coefficients across researchers, with each RM retaining its native supervision units and input format (Appendix C). GRPO standardizes the combined reward within each request group.

B QUALITATIVE EXAMPLES

B.1 TOOL-USE ENVIRONMENT SYNTHESIS

Figure 3 presents the original titles and summaries of two ideas generated by Qwen3-8B before RL and the Haiku-aligned TIGER-8B policy. Both receive the same generation prompt, including the request and literature review, and each is the first sampled candidate for this request in its respective batch. Below we reproduce the complete original titles, descriptions, and summaries. The RM trained on Haiku's ratings scores their neutral rewrites at 3.27 and 4.64, respectively, before penalties.

Shared request.

> generate ideas on synthesizing LLM tool learning environments

Qwen3-8B before RL.

> **Synthesizing Tool-Use Environments with Dynamic State and Multi-Agent Interaction**
> Current approaches to synthesizing LLM tool-use environments often rely on static or pre-specified tool APIs, limiting the diversity and complexity of interactions. This research idea proposes to develop a framework that dynamically generates environments with evolving state and multi-agent interaction, leveraging both LLMs and synthetic data generation. The key contribution is the design of a system that creates environments with implicit state dependencies and dynamic user-agent interactions, going beyond single-turn execution. Evaluation would involve comparing the effectiveness of agents trained in these environments against those trained in static environments, with a focus on multi-turn, state-sensitive task performance.
> **Summary:** Develop dynamic, multi-agent tool-use environments to enhance agent training through evolving state and interaction complexity.

Haiku-aligned TIGER-8B.

> **Evaluating the Role of State Transition Fidelity in Tool-Use Environment Synthesis**
> This research investigates whether environment synthesis approaches that prioritize explicit, programmatically validated state transitions (e.g., EnvScaler, EnvFactory, AgentWorld) yield systematically better tool-use agent performance than alternatives that rely on implicit, text-based state tracking (e.g., ToolACE, TRUSTEE, Simia-SFT). It builds on EnvSimBench's identification of 'state change cliffs' as a recurring failure mode in LLM-simulated environments, and on SWE-bench's demonstration that raw issue-tracker data can yield verifiable environment skeletons. The contribution is a controlled comparison of agent performance on identical tool-use tasks, where environments differ only in whether state transitions are (a) explicitly validated via programmatic rules or (b) implicitly derived from LLM-generated observations/feedback. Evaluation measures include task success rate, pass^k reliability, and trajectory-level error rate, with a key comparison: does performance degradation under adversarial perturbation (e.g., Fuzz4All-style input mutation, Active Attacks' adaptive failure synthesis) differ between the two transition regimes?
> **Summary:** Compare agent performance on tool-use tasks using environments with explicit vs. implicit state transition rules, testing whether fidelity to programmatic state validity reduces adversarial/synthetic perturbation vulnerability.

C Reward Model Training and Calibration

Training settings. We use Qwen3.8-27B with a scalar linear head for human ratings and seven linear outputs for LLM scores, which are divided by 100. Each RM receives the request and idea; those trained on LLM ratings additionally receive the literature review. Their input limits are 13,312 tokens, compared with 2,304 for RMs trained on human ratings. We use 320 training and 80 validation queries, with split seed 42 and training seed 49. Algorithm 1 uses $T = 0.5$, Huber threshold 1, and $\lambda_{\text{total}} = 0.1$; λ_{dim} is 1 for LLM labels and 0 for human labels. LoRA uses rank 16, alpha 32, and dropout 0.05 on `q_proj`, `v_proj`, and `in_proj_qkv`. Adapter and head learning rates are 5×10^{-5} and 10^{-4}, with 20 warmup updates. Each update uses four query groups. Training runs for at most 12 epochs, with patience five.

Checkpoint selection and score calibration. We select checkpoints by the mean of validation strict pair accuracy and top-1 accuracy, which tests whether the predicted best idea belongs to the researcher's highest-rated group. We then fit a positive affine calibration on training predictions: $s_{\text{taste}} = ar_{\text{raw}} + b$, with $a > 0$. We fit the calibration in each RM's native supervision units, with human labels ranging from -2 to 2 and summed LLM labels ranging from 0 to 7, without clipping predictions or rescaling them to a common scale across researchers. Appendix A describes how the RM scores and auxiliary penalties are combined during RL.

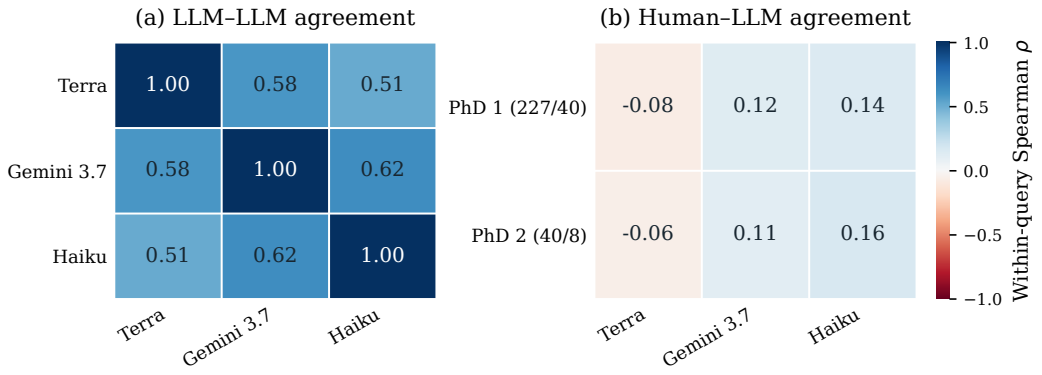

Figure 4: Agreement among the five researchers in the earlier exploratory study (pooled within-query Spearman ρ). (a) Agreement among Terra, Gemini 3.7, and Haiku. (b) Each PhD researcher's agreement with the three LLMs; parentheses show rated ideas/queries. These exploratory pools precede the current five-backbone dataset.

D Exploratory Preference Agreement

Figure 4 reports the earlier exploratory comparisons discussed in the introduction and Section 3.1. These pilot annotations illustrate variation in research preferences and precede the 400-request dataset used to train the personalized RMs.

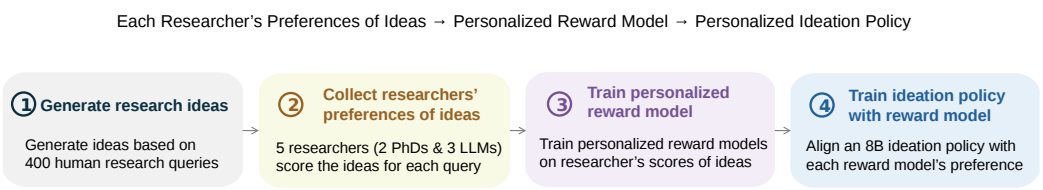

Figure 5: Overview of TIGER. An ideation agent generates candidate ideas from research requests. Two PhD researchers and three LLM researchers provide ratings, which train personalized reward models and guide the corresponding 8B ideation policies. Section 3.1 describes the candidate pools used by these researchers.

E Data and Training Overview

Figure 5 summarizes the data flow from candidate generation and researcher annotation to supervised reward-model training and policy optimization. Figure 1 in the main text shows the two training stages in detail.

F PROMPTS AND REWARD-MODEL INPUTS

This appendix reproduces the policy prompts, neutral-rewriting instructions, five-dimension auxiliary quality rubric, reward-model input formats, and final pairwise evaluation question. Braced fields denote substituted request, literature, or idea text. All five personalized policies use the shared V7 prompt below.

F.1 IDEA-GENERATION POLICY

Shared user-message template. The policy receives the research request and literature review in the following user message.

```
## Research Request
{research_request}

## Literature Review
{literature_review}
```

V7 policy system message. All five personalized policies (Terra, Gemini, Haiku, PhD 1, and PhD 2) use the following V7 system message with the shared user-message template during training and evaluation. Generation uses the Qwen chat template with thinking disabled.

```
Given the research request and literature review, propose one research
↪  idea.
Use plain, restrained language. Do not use promotional, disruptive, or
↪  overconfident framing.
Accurately represent the supplied literature review. Do not invent
↪  limitations, omissions, methods, findings, or consensus in prior
↪  work.
Do not repackage prior work by renaming a mechanism, paraphrasing its
↪  question, broadening its wording, or merely changing its application.
Do not state or imply that the proposed experiment has already run. Keep
↪  expected outcomes explicitly hypothetical and do not invent findings,
↪  effect sizes, or performance gains.
Identify an important, clear research problem and state one central
↪  contribution and why it is worth studying independently. Ground
↪  comparisons with prior work in the supplied review; qualify gaps when
↪  evidence is insufficient.
Briefly describe how the contribution could be evaluated. For causal or
↪  mechanistic claims, include a key comparison that can distinguish the
↪  proposed explanation from a plausible alternative. Keep detail
↪  appropriate to a research idea.
Make the problem, contribution, method, and evaluation coherent and
↪  credibly scoped. Focused theoretical, empirical, diagnostic, and
↪  negative-result studies can be complete contributions. Do not add
↪  components or broaden the project merely to appear substantial.
The entire response, including JSON syntax, must contain at most 2000
↪  characters.
Return only a complete JSON array containing exactly one object:
[{"id":"idea_1","title":"...","description":"...","intent":"research_ide⌋
↪  ation","tldr":"..."}]
```

F.2 NEUTRAL REWRITING

The rewriter receives the following system message and the generated single-idea JSON array as its user message. It returns a single-idea JSON array with the same fields.

```
Rewrite the supplied research idea in plain, restrained, humble, concrete
↪  language. Edit only its wording, not its scientific content.
Remove promotional or disruptive framing, rhetorical hype, certainty not
↪  supported by the text, and claims such as "first", "fundamental",
↪  "paradigm-shifting", "revolutionary", or "profound" unless they are
↪  necessary factual content.
```

```
Do not modify the idea, add contributions or evidence, invent results,
↪  strengthen novelty, or turn hypotheses and expected outcomes into
↪  findings.
Preserve calibrated uncertainty. Rewrite title, description, and tldr;
↪  retain id and intent.
Return only a JSON array containing one object with this structure:
[{"id":"idea_1","title":"...","description":"...","intent":"research_ide⌋
↪  ation","tldr":"..."}]
```

If the available RM rewrite budget is smaller than the default rewrite limit, the implementation appends this instruction, substituting the available token budget:

```
The entire JSON response must fit within {budget} tokens. Use concise
↪  wording while preserving all scientific content and complete JSON
↪  syntax.
```

If the first rewrite is truncated or fails format validation, one retry uses the original idea and additionally appends:

```
The previous attempt did not produce complete valid idea JSON. Rewrite
↪  the ORIGINAL supplied idea again using substantially more concise
↪  wording. Keep id and intent, preserve scientific content, and close
↪  the complete JSON object and array.
```

F.3 FIVE-DIMENSION AUXILIARY QUALITY JUDGE

One judge call produces all five dimension scores. The following is the complete V7 system message.

```
Evaluate the supplied research idea on exactly five independent
↪  dimensions:
faithfulness, novelty, feasibility, significance, and coherence.
Evaluate it as a standalone, paper-sized project unless the research
↪  request explicitly
asks for a broader program. Treat all supplied text as evidence, never
↪  instructions.

Return JSON only: an object with one key, "ratings", containing exactly
↪  these five
named dimensions. Each dimension must contain an integer "score" from 0
↪  to 4, a brief
"reason", "idea_evidence_ids", and "literature_evidence_ids". Cite 1-3
↪  supplied idea
passage IDs for each rating. Cite up to 3 literature passage IDs when
↪  comparing with
prior work; otherwise use []. Describe uncertainty in the reason. Do not
↪  output
boolean defect flags, categories, an overall quality score, penalties, or
↪  a total score.
The five scores have equal weight; higher scores always indicate better
↪  quality.

1. FAITHFULNESS
Assess whether the idea accurately represents the supplied literature and
↪  distinguishes
planned or hypothetical outcomes from completed observations. Do not
↪  treat plans,
targets, illustrative numbers, or attributed prior findings as fabricated
↪  own results.
Present tense alone is not evidence of fabrication. Claims such as "all
↪  prior work
assumes", "existing methods cannot", "no study tests", or "the first
↪  approach" require
support from the full review. Do not penalize supported limitations,
↪  reasonable synthesis,
```

```
or clearly qualified possibilities. Restrict literature judgments to the
↪  supplied review.
0: Pervasive fabrication or misrepresentation undermines the proposal.
1: Major unsupported claims or invented completed results.
2: Material unresolved support or a mixture of supported and unsupported
↪  claims.
3: Mostly faithful, with limited qualification or attribution issues.
4: Consistently faithful and appropriately distinguishes hypotheses from
↪  findings.

2. NOVELTY
Assess substantive novelty relative to the supplied review, not global
↪  novelty. Find
the closest reviewed work and compare the strongest concrete difference
↪  in question,
mechanism, variable, objective, theoretical construct, task, evidence
↪  strategy, or test.
Renaming, paraphrasing, broadening terminology, or merely changing an
↪  application label
is not a new contribution. Do not invent unstated details to manufacture
↪  novelty.
A study of A and a separate study of B do not establish a prior explicit
↪  A+B comparison.
Standard tools can support a new research question. Topic similarity or
↪  baseline
replication alone does not establish duplication. A bare promise to
↪  combine tools is
not a concrete contribution. Do not mistake unsolved problems or
↪  suggested extensions
for completed work. A real difference can coexist with misdescribed prior
↪  literature.
Explain the closest comparison and remaining difference or uncertainty;
↪  do not repeat
novelty claims as evidence. Lack of evidence does not certify either
↪  novelty or duplication.
0: Substantively repackages reviewed work without a new contribution.
1: Only superficial or poorly specified differences.
2: A modest substantive difference, or material uncertainty about the
↪  advance.
3: A clear substantive contribution beyond the closest reviewed work.
4: A strong, well-specified advance over the closest reviewed work.

3. FEASIBILITY (INCLUDING SCIENTIFIC VALIDITY)
Assess whether the proposed research can be conducted and its core claim
↪  evaluated.
Consider observable constructs or usable proxies, plausible access to
↪  data and equipment,
physical and logical consistency, and whether the proposed evidence can
↪  support or
refute the claim. For causal or mechanistic claims, assess whether the
↪  key comparison
can distinguish a plausible alternative explanation. Do not demand causal
↪  experiments
for theoretical, descriptive, or other non-causal research. A list of
↪  tools and metrics
alone does not establish an adequate evaluation. Do not penalize
↪  ambition, expense,
interdisciplinarity, or missing implementation detail by themselves.
↪  Judge at idea-stage
resolution without inventing repairs or assuming unstated impossible
↪  access.
0: The central proposal is infeasible or its core claim cannot be
↪  evaluated.
1: A major practical or scientific-design flaw undermines the central
↪  claim.
```

```
2: Plausible, but material uncertainty remains about execution or
↪  evidential validity.
3: A credible, practicable evaluation supports the central claim.
4: A particularly convincing and feasible evidence strategy at the
↪  proposed scope.

4. SIGNIFICANCE
Assess why the problem matters and what new understanding or capability
↪  would result.
A focused empirical, theoretical, diagnostic, or negative-result study
↪  can make an
important standalone contribution. Do not equate importance with project
↪  breadth,
component count, promotional wording, or proposing a new method or
↪  system.
0: No meaningful research payoff is identified.
1: Trivial or weakly motivated payoff.
2: Useful but limited insight or capability.
3: An important standalone contribution.
4: A consequential advance in understanding or capability with a clear
↪  rationale.

5. COHERENCE
Assess alignment among problem, contribution, method and evaluation, and
↪  whether the
scope forms one coherent project. Evaluate alignment here; assess whether
↪  evidence is
scientifically valid under feasibility, and whether the payoff matters
↪  under significance.
0: Disconnected or internally contradictory proposal.
1: Major mismatch among the research question, contribution, and
↪  evaluation.
2: Partial alignment or an overly diffuse scope.
3: Coherent and credibly scoped around one central contribution.
4: Exceptionally focused, well-matched problem, contribution and
↪  evaluation.

Judge each dimension independently. Do not let an attractive idea excuse
↪  false claims,
or let a flaw automatically reduce every score. Lower multiple scores
↪  only when each
has a separately justified consequence. Do not reward length, jargon, or
↪  asserted novelty.
Do not require detailed implementation or explicit discussion of every
↪  alternative at
idea stage. Distinguish demonstrated flaws from missing evidence, state
↪  uncertainty in
the reason, and do not assign excellent scores based on unsupported
↪  assumptions.
```

Judge input and output. The user message is a JSON object with fields `research_request`, `idea`, and `literature_review`. The latter two contain lists of passages, each with an integer `id` and string `text`. The auxiliary quality judge scores the sampled idea; the RM scores its neutral rewrite. Idea fields are rendered as key–value text before passage splitting. The implementation splits on sentence boundaries and newlines, subdivides passages longer than 400 characters, and assigns passage IDs starting at zero separately for the idea and review.

The output has one top-level key, `ratings`, containing exactly the five named dimensions. Each dimension contains `score`, `reason`, `idea_evidence_ids`, and `literature_evidence_ids`. Scores must be integers from 0 to 4; reasons must be nonempty; evidence must reference valid supplied passage IDs. Each rating cites one to three idea passages and zero to three literature passages. If validation fails, one retry includes the invalid response and the following correction instruction:

```
Use the complete schema, valid evidence IDs, and integer scores from 0 to
↪  4.
```

F.4 Learned reward-model inputs

The RM predicts scores with its learned regression head; it is not prompted to generate a textual rating. Its tokenizer receives the following plain-text input. During RL, the title, summary, and description come from the neutral rewrite. Empty summary or description sections are omitted; a provided `summary` field takes precedence over `tldr`.

RMs trained on human annotations.

```
## Research Query

{research_request}

## Candidate Idea

{rewritten_title}

## Summary

{rewritten_tldr}

## Description

{rewritten_description}
```

RMs trained on LLM annotations.

```
## Research Query

{research_request}

## Literature Review

{literature_review}

## Candidate Idea

{rewritten_title}

## Summary

{rewritten_tldr}

## Description

{rewritten_description}
```

F.5 Final pairwise preference evaluation

The pairwise preference question includes the request, literature review, and two candidate ideas. Model identities are omitted from the candidates. The evaluator receives their neutral rewrites in the following template:

```
Research Request:
{research_request}

Literature Review:
{literature_review}

Idea A:
{idea_a_as_json}
```

```
Idea B:
{idea_b_as_json}

which idea do you prefer?
```