CONSTRUCTIVE DIVERGENCE: A JUDGE-FREE REWARD FOR CREATIVE LANGUAGE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Creativity is a core capability for autonomous research agents, yet optimizing it remains challenging: verifiable reward signals are scarce, human annotations and LLM-as-a-judge can be costly or contaminated, and heuristic rewards are vulnerable to hacking. To address this challenge, we propose Constructive Divergence, a scalable creativity score requiring only the token probabilities of two reference models. Grounded in the psychological view that creativity requires both novelty and value, we operationalize novelty as the surprisal of an output and value as the recoverability of its original intent. Concretely, we compute surprisal as the negative log-likelihood of the output under a strong reference model, and recoverability as the conditional log-likelihood of the prompt given the output under a second reference model. Notably, Constructive Divergence serves as both a reliable evaluation metric and a cost-effective training reward. As an evaluator, Constructive Divergence is deterministic and robust to reference-model choice, with 70% agreement across model configurations. On paper-abstract generation, optimizing models with DPO and GRPO using Constructive Divergence improves creativity with an average increase of 25%. Our trained models achieve creativity gains up to 4.0 times larger than the average gain of the baselines, and their outputs are preferred by LLM-as-a-judge models in up to 61% of pairwise comparisons, as further examined with human annotators.

Autonomous research agents now explore new ideas rather than merely executing them (Lu et al., 2024; Si et al., 2025). Such agents therefore need **creativity**, but defining, evaluating, and optimizing it remain challenging because, unlike mathematical reasoning and coding tasks, where verifiable rewards have driven progress (DeepSeek-AI et al., 2025; Lambert et al., 2025), creative ideation has no verifiable answer. Ideas exhibit varying degrees of novelty and value, rather than a binary distinction between right and wrong.

To measure or improve creativity, prior work has relied on human annotation (Chakrabarty et al., 2024; Si et al., 2025; Gómez-Rodríguez & Williams, 2023), LLM-as-a-judge (Zheng et al., 2023), or heuristic rewards (Li et al., 2016; Zhu et al., 2018; Olson et al., 2021; Beaty & Johnson, 2021). Human studies, however, are costly and limited in scale (Ismayilzada et al., 2025). LLM-as-a-judge scales better than human annotation, but it is vulnerable to contamination (Sainz et al., 2023; Golchin & Surdeanu, 2024) and familiarity bias (Stureborg et al., 2024). In our preliminary study, LLM-as-a-judge models recall the title of a published abstract in up to 61% of cases, and they strongly prefer abstracts they recognize to those they do not. They also exhibit presentation-order bias, including among the strongest models (Zheng et al., 2023; Wang et al., 2024). Finally, heuristic diversity metrics based on lexical overlap (Papineni et al., 2002; Lin, 2004; Zhu et al., 2018) or embedding distance (Reimers & Gurevych, 2019; Beaty & Johnson, 2021) are vulnerable to reward hacking (Skalse et al., 2022; Gao et al., 2023; Pan et al., 2022; Pang et al., 2023) and may produce degenerate text when directly optimized (Pang et al., 2023; Paulus et al., 2018).

In this paper, We adopt a psychological definition of creativity: a creative product must be both novel and effective (Boden, 2004; Runco & Jaeger, 2012). To operationalize these two conditions, we use two proxies: *surprisal* for novelty and *recoverability* for effectiveness, where an effective output should remain appropriate to the stated intent (Runco & Jaeger, 2012). More specifically, surprisal assigns higher scores to outputs that depart from the model's learned expectations, and in expectation under the policy, it decomposes into the policy entropy plus the KL divergence from the reference model. Recoverability assigns higher scores to outputs from which the original intent can

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.

be recovered, and is computed as the mean token log-probability of the original prompt conditioned on the generated output.

Optimizing either score in isolation leads to failure, because maximizing surprisal alone drives the output away from the stated intent, whereas maximizing recoverability alone leads to restating the prompt. Therefore, we combine these two scores into a single objective and call it **Constructive Divergence**, as it rewards informative departures from the reference model's expectations that preserve the original intent.

We use Constructive Divergence to train language models with DPO (Rafailov et al., 2023) and GRPO (Shao et al., 2024) on a research-idea exploration task. As a result, training increases Constructive Divergence itself by 25% on average and raises it on 129 (DPO) and 130 (GRPO) of the 130 held-out prompts; three independent LLM-as-a-judge models prefer outputs from the trained model in up to 61% of pairwise comparisons, as further examined with human annotators. The surprisal and recoverability gains are reproduced when the held-out outputs are re-scored with seven alternative models for each of the two reference models, and the trained policies move away from the nearest published abstract, whereas a policy trained with an LLM-as-a-judge reward moves closer. Ablations confirm that each component is necessary: removing recoverability lowers recoverability below the base policy, removing surprisal leaves surprisal nearly unchanged, and removing paraphrase correction raises perplexity. Furthermore, when used as a quantitative metric for the same task, Constructive Divergence is substantially less sensitive than LLM-as-a-judge to output length (Hu et al., 2025), sampling temperature (Stureborg et al., 2024), and memorization (Sainz et al., 2023; Golchin & Surdeanu, 2024); on the same model–model pairs, it prefers the longer abstract in 52.2% of cases (not significant), whereas GPT-4o (OpenAI et al., 2024) and GPT-5.2 (OpenAI, 2025a) do so in 55.8% and 63.2%. Moreover, our score yields significantly more robust evaluations than LLM-as-a-judge evaluations, as it is derived from token probabilities of frozen models.

Our contributions are threefold.

- We introduce Constructive Divergence, a psychology-grounded creativity score computed from the token probabilities of two frozen models, without a single label.
- Training with Constructive Divergence using DPO and GRPO improves model creativity under LLM-as-a-judge evaluation, as further examined with human annotators, and moves outputs away from existing literature; ablations show that each component of the score is necessary.
- Constructive Divergence is a model-agnostic evaluator whose gains are reproduced under seven alternative models for each reference model, with substantial agreement across reference models and reduced vulnerability to memorization and other biases compared with LLM-as-a-judge evaluations.

1 Related Work

Measuring creativity. Psychology defines creativity through novelty and value (Boden, 2004; Runco & Jaeger, 2012). Language-model metrics measure how well a text can be reconstructed from web snippets (Lu et al., 2025) or how much outputs vary (Zhang et al., 2025; Jiang et al., 2025), but they often disagree with one another and with human judgments (Lu et al., 2026), transfer poorly across domains (Hou et al., 2026), or reduce to n-gram diversity (Saakyan et al., 2026). Semantic entropy and distributional surprise (Tan et al., 2026; Bunescu & Uduehi, 2022), used on their own, overlook task relevance and can assign high scores to valueless outputs. Constructive Divergence instead rewards content novelty together with preservation of the original intent.

Research ideation with language models. Ideation systems promote novelty at inference time through literature search, tournaments, and quality–diversity search (Li et al., 2025; Hu et al., 2024; Gottweis et al., 2026), often within end-to-end research agents (Lu et al., 2024; Yamada et al., 2025), or at training time with LLM reward models, debate judges, or learned reviewers (Li et al., 2026; Salimi et al., 2026; Weng et al., 2025). The resulting ideas are often less feasible (Si et al., 2025), lose their advantage upon execution (Si et al., 2026), reuse prior work (Gupta & Pruthi, 2025), and cover less of the idea space (Chen et al., 2026). Our reward is computed from the token probabilities

of frozen models, avoiding the cost of LLM-as-a-judge and the mismatch between training and inference.

Verifier-free rewards and probability signals. Beyond verifiable rewards (DeepSeek-AI et al., 2025; Lambert et al., 2025), verifier-free methods reward reference-answer probability (Zhou et al., 2026; Yu et al., 2025b), reasoning perplexity (Liu et al., 2025), or self-confidence (Zhao et al., 2026). LLM-as-a-judge and learned reward models offer another reward for open-ended writing (Zheng et al., 2023; Jia et al., 2025; Liao et al., 2026; Wu et al., 2026), but are sensitive to presentation order (Wang et al., 2024), self- or familiarity preference (Panickssery et al., 2024; Wataoka et al., 2024), contamination (Golchin & Surdeanu, 2024), and reward hacking (Skalse et al., 2022; Gao et al., 2023). Probability signals have mainly served decoding, by contrasting models (Li et al., 2023), targeting information content (Meister et al., 2023; Basu et al., 2021), or penalizing previously generated branches (Park et al., 2025; 2026); Constructive Divergence turns them into a training reward that does not rely on LLM-as-a-judge.

2 METHODOLOGY

2.1 TASK DEFINITION

For autonomous research agents, exploring multiple candidate ideas improves the novelty and diversity of generated ideas (Hu et al., 2024) and is central to agentic research pipelines (Yamada et al., 2025). In this paper, we study how to promote creativity in research agents specifically at this ideation stage. To instantiate this problem, we define a new task: research-abstract generation from author-provided keywords. The keywords specify the topic and central concepts of the intended research direction without prescribing a particular problem or solution. Generating an abstract requires the policy to resolve this open-ended intent into a concrete research idea that connects a motivation, a proposed method, and its anticipated contribution. This task directly tests whether an agent can turn a research direction into a concrete, novel, and valuable proposal, and it can be readily incorporated into existing research-agent pipelines (Lu et al., 2024; Yamada et al., 2025). It also isolates ideation from confounding stages such as literature retrieval (Hu et al., 2024; Mitchener et al., 2025), implementation (Lu et al., 2024), and experimental execution (Lu et al., 2024; Yamada et al., 2025; Feng et al., 2026).

Formally, let $q \sim \mathcal{Q}$ denote the input prompt, which combines a fixed instruction to generate a research abstract with a set of keywords specifying the research intent, and let $x \sim \pi_\theta(\cdot \mid q)$ denote an abstract generated by a policy π_θ. Following the standard psychological definition of creativity (Boden, 2004; Runco & Jaeger, 2012), we aim to train models to generate abstracts that exhibit both *novelty* and *recoverability*. We seek a sample-level creativity score $\mathrm{C}(x, q)$ that captures these two requirements and can serve both as an evaluator and as an optimization objective:

$$\max_\theta \ \mathbb{E}_{q\sim\mathcal{Q},\, x\sim\pi_\theta(\cdot|q)}\big[\mathrm{C}(x, q)\big]. \tag{1}$$

In the following sections, we operationalize novelty and recoverability, which constitute $\mathrm{C}(x, q)$.

2.2 NOVELTY SCORE

We operationalize novelty as *surprisal*, which is the negative log-probability of an output under a frozen reference language model V:

$$S_{\mathrm{raw}}(x \mid q) = -\log p_V(x \mid q). \tag{2}$$

Intuitively, content that departs from the reference model's learned expectations may contain novel ideas. Surprisal turns this intuition into a sample-level signal by assigning a higher score to an output that V considers less probable given the same q. Note that this interpretation relies on V being sufficiently capable and well-calibrated for the target domain, such that low probability primarily reflects a meaningful departure from learned regularities.

Distributional motivation. We next examine the expectation of raw surprisal under outputs sampled from $\pi_\theta(\cdot \mid q)$ to understand the policy-level effect of optimizing it:

$$\mathbb{E}_{x\sim\pi_\theta(\cdot|q)}[S_{\mathrm{raw}}(x \mid q)] = H(\pi_\theta(\cdot \mid q)) + D_{\mathrm{KL}}(\pi_\theta(\cdot \mid q) \,\|\, p_V(\cdot \mid q)). \tag{3}$$

Appendix A.1.1 provides the full derivation of this identity. This identity exposes two ways in which expected surprisal can increase: greater entropy spreads probability across multiple possible outputs, while greater KL divergence moves the policy away from the reference model's predictive distribution.

Sentence-level decomposition. We use sentences, rather than individual tokens, as the units of novelty because token probabilities are highly sensitive to lexical choice. For example, our analysis in Appendix B.7 shows that the highest-scoring tokens are often semantically uninformative words, such as "these", "by" and "and." We therefore aggregate sentence-level surprisals into a single novelty score for each output. However, sentence-level surprisal is still sensitive to output length, local predictability, and unusual wording. We therefore refine it through token normalization, context calibration, and paraphrase correction in the following steps.

Token and context normalization. Motivated by the entropy term in Equation 3, we account for the fact that raw surprisal can reflect context-dependent uncertainty as well as novelty. For example, after a broad preceding context such as "We study how to improve LLMs," even a conventional continuation may receive low probability not because it is novel, but because many methods remain plausible. We therefore calibrate each sentence's surprisal against the typical predictability of continuations under V in the same context.

Let $a = (a_1, \ldots, a_m)$ denote a tokenized text span, where a_j is its j-th token, m is the number of tokens in a, and $a_{<j} = (a_1, \ldots, a_{j-1})$ is the prefix preceding a_j. Given a conditioning context c, we define the mean token log-probability of a under a model M as

$$\ell_M(a \mid c) = \frac{1}{m} \sum_{j=1}^{m} \log p_M(a_j \mid c, a_{<j}). \tag{4}$$

We split x into sentences $s_1, \ldots, s_n$ and let $c_i = (q, s_{<i})$ be the context preceding sentence s_i. Since the intrinsic predictability of continuations varies across contexts, we estimate a context-specific baseline using a Monte Carlo average over K sentence-level continuations $y_{i1}, \ldots, y_{iK} \sim p_V(\cdot \mid c_i)$:

$$\widehat{\mu}_V(c_i) = \frac{1}{K} \sum_{k=1}^{K} \ell_V(y_{ik} \mid c_i). \tag{5}$$

We then define the contextual surprisal of s_i as

$$\widehat{\mu}_V(c_i) - \ell_V(s_i \mid c_i).$$

It is positive when s_i is less predictable than continuations that V would typically produce in the same context. Thus, the sampled baseline calibrates surprisal to local context rather than treating every position as equally difficult.

Paraphrase correction. Even after context normalization, a semantically ordinary sentence can receive a high score because of rare lexical choices or irregular syntax. To reduce this surface-level contribution, we replace each sentence with a semantics-preserving paraphrase, $\tilde{s}_i = P(s_i)$. We generate $\tilde{s}_i$ using V, conditioned on s_i and a fixed rewriting instruction, and score it under the original context c_i. Using V for both paraphrasing and scoring controls for the reference model's own lexical and syntactic preferences. Our final novelty term is

$$S(x \mid q) = \frac{1}{n} \sum_{i=1}^{n} \left[\widehat{\mu}_V(c_i) - \ell_V(\tilde{s}_i \mid c_i)\right]. \tag{6}$$

We verify that this paraphrasing does not alter what a sentence claims in Appendix A.3.

Although Equation 6 is not algebraically equivalent to Equation 3, it operationalizes the same underlying intuition: novelty corresponds to content that is unexpected under V, while correcting for contextual uncertainty, sentence length, and surface form.

2.3 RECOVERABILITY SCORE

Surprisal alone cannot distinguish a novel proposal from irrelevant or nonsensical text. We therefore measure whether the original input prompt q remains recoverable from the output x. Let $t(q) = (t_1(q), \ldots, t_m(q))$ denote the tokenized serialization of the keyword component of q, and let ρ denote a fixed instruction asking the model to recover these keywords from an abstract. We define recoverability as

$$R(x \mid q) = \ell_D(t(q) \mid \rho, x) = \frac{1}{m} \sum_{j=1}^{m} \log p_D\left(t_j(q) \mid \rho, x, t_{<j}(q)\right), \tag{7}$$

where D is another strong reference model. An output receives a high recoverability score when D assigns high probability to $t(q)$ after conditioning on the recovery instruction ρ and the output x. Thus, a surprising output that is unrelated to the input topic or nonsensical receives low recoverability even if it remains fluent.

Recoverability also admits an information-theoretic interpretation. Let $Q \sim \mathcal{Q}$ and $X \sim \pi_\theta(\cdot \mid Q)$, and write $T(Q) = t(Q)$ for the keyword serialization. Since the instruction ρ is fixed,

$$I(T(Q); X) \geq H(T(Q)) + \mathbb{E}_{\substack{Q \sim \mathcal{Q} \\ X \sim \pi_\theta(\cdot \mid Q)}} \left[\log p_D\left(T(Q) \mid \rho, X\right)\right]. \tag{8}$$

Thus, maximizing expected recoverability encourages outputs to retain information about their conditioning keyword intent.

2.4 CONSTRUCTIVE DIVERGENCE

Combining novelty and recoverability. To combine the two terms, *novelty* and *recoverability*, we normalize them by fixed standard deviations σ_S and σ_R estimated on a calibration set of candidate outputs. Then we define the final score C:

$$\mathrm{C}(x, q) = \frac{S(x \mid q)}{\sigma_S} + \frac{R(x \mid q)}{\sigma_R}. \tag{9}$$

Here, σ_S and σ_R are estimated before evaluation and thereafter held fixed. The additive score rewards novelty while penalizing poor intent preservation. We call this combined objective *Constructive Divergence*, as it favors divergence from V that remains constructive with respect to the input intent.

Distinct reference models. At this point, the two reference models, V and D, should not be derived from the same joint distribution. Suppose that $p_V(x \mid q)$ and $p_D(q \mid x)$ were exact Bayesian inverses under a common joint distribution. Then, for the unnormalized sum,

$$-\log p_V(x \mid q) + \log p_D(q \mid x) = -\log p_V(x) + \log p_V(q). \tag{10}$$

For a fixed q, $\log p_V(q)$ is constant. Thus, maximizing this expression rewards only high marginal surprisal, $-\log p_V(x)$, including noise. We therefore use independently trained models from different families for V and D to avoid this algebraic cancellation.

Safeguard filters. Although recoverability discourages topical drift and nonsensical outputs, it does not rule out every malformed-output strategy. We therefore accompany Constructive Divergence with safeguard filters for fluency and degeneration. The operational definitions and thresholds for each filter are provided in Appendix A.10.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Dataset and policy. We construct prompts from the author-provided keywords of accepted ICLR 2025 papers. We use 1,000 keyword sets for training and 130 disjoint sets for evaluation; the corresponding titles and abstracts are never shown to the policy. A fixed instruction asks the model to turn each keyword set into a one-paragraph research abstract. We use Gemma 4 12B-it (Gemma Team, 2026) as the policy and train rank-32 LoRA adapters. All methods start from the same base model.

Table 1: Evaluator diagnostics. CD, OS, LP, CP, TR and RΔ indicate Constructive Divergence, order sensitivity, longer-output preference, classic-paper preference, title recall, and recognition-associated preference changes, respectively. The judges behind each value of the LLM judges row are listed in Appendix A.4. Human denotes Constructive Divergence's win rate in the human preference study over 26 evaluator-disagreement pairs.

Evaluator	OS ↓	LP	CP ↓	TR ↓	RΔ ↓	Human ↑
CD	0.0%	46.7%	37.2%	0.0%	—	55.0%
LLM judges	30.0%	58.8%	76.7%	61.0%	+18.2 pp	45.0%

Reward models and calibration. We use frozen gpt-oss-120B (OpenAI, 2025b) as the reference model V and frozen Llama-3.1-70B-Instruct (Grattafiori et al., 2024) as the second reference model D. Appendix A.7 reports the reason for this choice of reference models. These models remain fixed throughout training and evaluation. For each sentence context, we draw $K = 8$ continuations from V to estimate the contextual baseline in Equation 5; paraphrases are produced by greedy decoding from the same V. We apply the fluency and degeneration filters described in Appendix A.10.

3.2 Constructive Divergence as an evaluator

Before using Constructive Divergence as a reward, we compare its bias-relevant properties with those of LLM-as-a-judge and conduct a human preference study with three machine-learning Ph.D. annotators on pairs for which our score and an LLM-as-a-judge model make conflicting predictions.

3.2.1 Comparison with LLM-as-a-judge

Table 1 summarizes several advantages of Constructive Divergence over LLM-as-a-judge. On average, three LLM-as-a-judge models, GPT-5.2 (OpenAI, 2025a), GPT-5.6 Sol (OpenAI, 2026b), and Gemini 3.6 Flash (Google, 2026), reverse their verdicts on 30.0% of pairs when the presentation order is swapped, and LLM-as-a-judge models favor longer abstracts. LLM-as-a-judge is also susceptible to familiarity: GPT-5.2 selects a classic foundational idea over a 2026 idea in 76.7% of order-consistent decisions, and contaminated judges disproportionately select papers they explicitly recognize. Constructive Divergence is substantially more robust to most of these weaknesses, as further examined with human annotators in Section 3.2.2. We also conduct cross-reference-model agreement evaluations in Appendix B.2.

3.2.2 Human judgments on disagreement pairs

To determine which evaluator better aligns with human judgments when they disagree, we construct a diagnostic set consisting only of pairs on which Constructive Divergence and GPT-5.2 select different outputs. GPT-5.2 receives both abstracts and their shared keywords and is asked which presents the more original idea, whereas Constructive Divergence selects the output with the higher independently computed score $\mathrm{C}(x, q)$. We control for length and presentation-order biases before asking human raters to compare both outputs in each retained pair. More details are in Appendix A.5.

As shown in Table 1, human raters prefer the independently scored higher-C output in 55.0% of pairs, compared with 45.0% for the output selected by GPT-5.2. Notably, GPT-5.2 prefers the published abstract to the model-generated output in every case, whereas human raters do so in only 60.0%, revealing a substantial disagreement between human judgments and those of GPT-5.2.

3.3 Constructive Divergence as a reward

3.3.1 Training setup

Training. We train separate policies with DPO (Rafailov et al., 2023) and GRPO (Shao et al., 2024). Detailed preference construction, objectives, safeguard filters, and training hyperparameters are provided in Appendix A.9.

Table 2: Creativity comparison across various reward functions and diversity-promotion methods. GRPO$_{CD}$ and DPO$_{CD}$ denote policies trained with GRPO and DPO, respectively, using Constructive Divergence. We bold the highest and underline the second-highest scores for S, R, and C.

(a) Automatic metrics

Method	Surprisal S	Recoverability R	Score C	PPL	Words
Base (Gemma 4 12B-it)	3.712	-2.645	7.094	6.33	176.7
GRPO$_{CD}$	<u>4.303</u>	-2.484	<u>8.869</u>	7.55	174.1
DPO$_{CD}$	4.165	-2.271	8.734	7.64	177.2
w/o Paraphrase	4.102	<u>-2.257</u>	8.577	8.43	175.3
w/o Recoverability	**4.399**	-2.660	**8.933**	8.12	174.7
w/o Surprisal	3.759	$\mathbf{-2.122}$	7.802	6.77	176.4
Creativity Instruction	3.741	-3.037	6.736	8.96	181.7
Self-Refinement	3.888	-3.015	7.158	11.52	180.2
DPO$_{LLM}$	3.684	-2.737	6.916	7.18	183.0
Best-of-8	4.124	-2.479	8.391	6.60	176.6
Best-of-16	4.227	-2.502	8.645	6.73	176.8
Base$_{t=1.5}$	3.847	-2.667	7.433	8.07	177.1
Base$_{t=1.8}$	3.898	-2.743	7.486	18.83	183.1

(b) Pairwise preference

Win Rate of First over Second	LLM	Human
GRPO$_{CD}$ / Base	56.4%	40.0%
DPO$_{CD}$ / Base	60.6%	54.5%
DPO$_{CD}$ / DPO$_{LLM}$	13.3%	—

Baselines. We compare our trained policies against five classes of baselines: (1) DPO$_{LLM}$, trained on preference rankings from an LLM-as-a-judge model; (2) Creativity Instruction, which uses an explicit creativity instruction; (3) Self-Refinement, which performs one-step self-refinement using the base policy; (4) Best-of-N selection, which selects from $N \in \{2, 4, 8, 16\}$ samples from the base policy using Constructive Divergence; and (5) Base$_{t=\ldots}$, which simply increases the base policy's sampling temperature from its default value of 0.9. The ablation variants use the same DPO training recipe as our method and differ only in their ranking functions: *w/o Paraphrase* removes paraphrase correction, *w/o Recoverability* removes the recoverability term, and *w/o Surprisal* removes the surprisal term. Except where otherwise noted, all methods use the same base policy and decoding settings. Further details appear in Appendix A.8.

Automatic evaluation metrics. Our automatic metrics are surprisal S, recoverability R, their calibrated sum Score C, perplexity, and output length. We compute sentence-averaged perplexity as $\exp[-F_D(x)]$, where $F_D(x)$ is the mean across sentences of their mean token log-probability under the second reference model D. For S, R, and C, we first average a policy's five samples for each prompt, yielding one score per policy–prompt pair.

Pairwise preference. We also conduct pairwise preference evaluations with both LLM-as-a-judge and human raters. Three LLM-as-a-judge models—Gemini 3.6 Flash, GPT-5.2, and GPT-5.6 Sol—evaluate 260 anonymized output pairs per comparison in both presentation orders; the pairs match the first two of the five samples of each policy for each of the 130 held-out prompts by sample index. We count a win only when the two orders yield the same directional preference, and report the win rate excluding ties, averaged over the three judges. For each model comparison, human raters evaluate the same 27 anonymized pairs on a five-point scale: "A is much more creative," "A is somewhat more creative," "similar," "B is somewhat more creative," and "B is much more creative." To mitigate length bias (Hu et al., 2025), we include only pairs whose lengths differ by at most 18 words. More details about the LLM-as-a-judge protocol are in Appendix B.8.

Table 3: Gemini 3.6 Flash-based analysis of outputs from each policy on 130 held-out prompts. *Recombination* is the share of ideas judged to be mainly a combination of existing components, and *specificity* is a 1–5 rating of how distinctive the stated idea is. The remaining columns show how often each rhetorical role contains the highest-scoring sentence, normalized by that role's overall frequency among sentences.

	(a) Idea coding		(b) Rhetorical-role concentration			
Method	Recombination	Specificity	Implication	Result	Mechanism	Contribution
Base	0.529	3.31	1.79	1.47	0.53	0.76
GRPO$_{CD}$	0.480	3.11	1.55	1.53	0.66	0.66
DPO$_{CD}$	0.448	3.33	1.60	1.41	0.72	0.75
DPO$_{LLM}$	0.600	3.86	2.74	1.05	0.51	0.74

3.3.2 Optimization results

As shown in Table 2(a), both DPO and GRPO with our Constructive Divergence as a reward raise the two terms, surprisal and recoverability together. GRPO increases surprisal more, whereas DPO increases recoverability more. The Creativity Instruction and Self-Refinement baselines barely move surprisal while cutting the recoverability term, which is exactly the failure the recoverability term exists to catch.

The ablation results show that dropping either term disrupts the balance that the term was designed to maintain. w/o Recoverability achieves the highest surprisal but lower recoverability than the base policy. Conversely, w/o Surprisal exhibits the best recoverability in the table, while its surprisal barely improves over that of the base policy. Finally, removing paraphrase correction increases perplexity, suggesting that the apparent surprisal gain arises from surface irregularity.

DPO$_{LLM}$ lowers both surprisal and recoverability, and Best-of-N raises C, but remains below the GRPO and DPO policy means. High-temperature sampling yields higher surprisal but lower recoverability, resulting in a lower C than the mean scores of the GRPO and DPO policies. In summary, only policies trained with Constructive Divergence as the reward achieve positive improvements in both surprisal and recoverability, raising C by 1.64 (DPO$_{CD}$) and 1.78 (GRPO$_{CD}$), 3.7 and 4.0 times the average gain of the baselines in Table 2(a) (0.44), showing that our metric captures a unique value that cannot be increased through prompting, sampling strategies, or alternative rewards alone. Full results are in Appendix B.6.

3.3.3 Human and LLM preference evaluation

As shown in Table 2(b), LLM-as-a-judge models prefer DPO and GRPO outputs to those of the base model. However, they strongly preferred outputs from the model trained with GPT-5.2 as a reward over those from Constructive Divergence. This suggests that LLM-as-a-judge may be biased toward outputs optimized for their own judgments, which we examine in the following analysis section.

For human evaluation, we take the majority vote of three machine-learning Ph.D. annotators on each of the 27 unique pairs per comparison, excluding ties and position-swapped repeats. As shown in Table 2(b), human raters prefer DPO$_{CD}$ over the base policy in 54.5% of pairs and GRPO$_{CD}$ over the base policy in 40.0% of pairs; neither comparison is statistically significant (sign test, $p = 0.83$ and 0.42). Given the low agreement among annotators, these results suggest that assessing the originality of research abstracts is difficult even for experts. This finding is consistent with prior work (Si et al., 2025). Further details of the human preference study are provided in Appendix A.6.

4 Analysis

Idea-shift analysis. As Table 3(a) shows, according to Gemini 3.6 Flash coding, training with Constructive Divergence moves ideas away from recombination for both DPO$_{CD}$ and GRPO$_{CD}$. However, DPO$_{CD}$ leaves specificity unchanged, while GRPO$_{CD}$ lowers it. The DPO$_{LLM}$ tends to create recombinative ideas even more often than the base model, while its ideas are rated the most specific.

Table 4: Embedding geometry of each policy's held-out outputs. *Inner-Diversity* is the mean pairwise cosine distance among the five samples a prompt produces. *Human-Distance* is the cosine distance from each output to the closest real ICLR abstracts, excluding the output's own source paper. C gain is the difference between each policy's C score and that of the base policy. $\text{DPO}_{CD,\,t=1.5}$ is the same policy as DPO_{CD}, except that its sampling temperature is increased to 1.5.

	Inner-Diversity	Human-Distance	C gain
Base	0.0793	0.2905	—
DPO_{CD}	0.0684	0.2968	+1.64
GRPO_{CD}	0.0805	0.2972	+1.78
DPO_{LLM}	0.0752	0.2851	−0.18
$\text{DPO}_{CD,t=1.5}$	0.0809	0.2993	+1.66
$\text{Base}_{t=1.5}$	0.1016	0.2907	+0.34

To analyze where creativity is concentrated, we ask Gemini 3.6 Flash to assign a rhetorical role to each sentence and then examine the role of the highest-scoring sentence. As shown in Table 3(b), GRPO_{CD} and DPO_{CD} increase the concentration of mechanism sentences relative to the base model, while GRPO_{CD} also slightly increases the concentration of result sentences. In contrast, both policies reduce the concentration of implication sentences. Contribution sentences are the most templated part of an abstract and are predicted well by the reference model; accordingly, they are unlikely to be a primary source of surprisal.

These results suggest that training with LLM-as-a-judge as the reward signal encourages the policy to generate more recombinative and more specific-sounding ideas, while concentrating the highest-scoring sentences in implications. This behavior is somewhat misaligned with the exploratory goal of early-stage research ideation. By contrast, our reward produces less recombinative ideas, maintains specificity, and shifts concentration toward mechanism sentences, thereby better aligning with exploratory research ideation.

Diversity analysis. To analyze the diversity of outputs, we embed each policy's output with OpenAI's text-embedding-3-small model (OpenAI, 2024b), ℓ_2 normalize the embeddings, and check the average within-prompt diversity and distance to its nearest human-written abstract, excluding the prompt's source paper.

As shown in Table 4, DPO_{CD} narrows average within-prompt diversity, but high-temperature sampling restores diversity to above the base model's level while retaining a substantially larger C gain than $\text{Base}_{t=1.5}$. GRPO_{CD} also widens average within-prompt diversity, whereas DPO_{LLM} does not. For distance to the nearest human-written abstract, all policies trained with Constructive Divergence as a reward show increased distance relative to the base model, whereas DPO_{LLM} decreases it. This suggests that using LLM-as-a-judge as a reward encourages the policy to generate abstracts that resemble existing literature rather than novel ideas. Further analyses including qualitative analysis are in Appendix B.

5 Conclusion

We introduced Constructive Divergence, a creativity score computed from the token probabilities of two frozen reference language models. It operationalizes novelty as surprisal and value as the recoverability of the original intent. As an evaluator, Constructive Divergence is deterministic, insensitive to presentation order and output length, and robust to memorization, as further examined with human annotators. As a reward, Constructive Divergence improves both surprisal and recoverability when used to train DPO and GRPO, while yielding outputs preferred by LLM-as-a-judge models and shifting ideas away from simple recombination and existing literature. Thus, Constructive Divergence serves as both a reliable evaluator and an effective reward signal for creativity enhancement, particularly for autonomous research agents.

AI USE STATEMENT

We utilized OpenAI's ChatGPT for limited assistance in refining the writing and formatting of this paper. All substantive contributions, including the core methodology, experiments, and analysis, were conducted independently by the authors. Our usage complies with OpenAI's Terms of Use and Usage Policies.

ETHICS STATEMENT

This work primarily focuses on measuring and improving the creativity of generative models and therefore does not directly raise ethical concerns. However, we acknowledge that generated research ideas may be mistaken for validated findings or may resemble prior work without attribution, so they should be verified by researchers before use. The human evaluation was conducted with Ph.D. students with machine-learning expertise, who participated voluntarily without compensation and compared anonymized abstracts only; no personal data about the raters is reported.

REPRODUCIBILITY STATEMENT

Section 2 defines every component of Constructive Divergence, and Section 3.1 specifies the data splits, policy, and reference models. Appendix A.2 gives the paraphrase instruction and sentence splitting, Appendix A.9 the DPO and GRPO hyperparameters, Appendix A.10 the safeguard thresholds, Appendix A.8 the prompts and decoding settings of all baselines, and Appendix B.8 the LLM-as-a-judge and human-evaluation protocols. Code for scoring, training, and evaluation is available at `https://anonymous.4open.science/r/2026_iclr_submission-27B5/`.

REFERENCES

Anthropic. System Card: Claude Opus 5. `https://www.anthropic.com/claude-opus-5-system-card`, July 2026.

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *International Conference on Learning Representations*, 2021.

Roger E. Beaty and Dan R. Johnson. Automating creativity assessment with SemDis: An open platform for computing semantic distance. *Behavior Research Methods*, 53(2):757–780, 2021. doi: 10.3758/s13428-020-01453-w.

Margaret A. Boden. *The Creative Mind: Myths and Mechanisms*. Routledge, 2nd edition, 2004.

Razvan C. Bunescu and Oseremen O. Uduehi. Distribution-based measures of surprise for creative language: Experiments with humor and metaphor. In *Proceedings of the 3rd Workshop on Figurative Language Processing*, pp. 68–78, 2022. doi: 10.18653/v1/2022.flp-1.10.

Tuhin Chakrabarty, Philippe Laban, Divyansh Agarwal, Smaranda Muresan, and Chien-Sheng Wu. Art or artifice? large language models and the false promise of creativity. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 2024.

Ziyu Chen, Yilun Zhao, and Arman Cohan. Measuring the gap between human and LLM research ideas. *arXiv preprint arXiv:2607.01233*, 2026.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025.

Shiyang Feng, Runmin Ma, Xiangchao Yan, Yue Fan, Yusong Hu, et al. InternAgent-1.5: A unified agentic framework for long-horizon autonomous scientific discovery. *arXiv preprint arXiv:2602.08990*, 2026.

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, 2023.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. doi: 10.48550/arXiv.2607.02770. URL https://arxiv.org/abs/2607.02770.

Shahriar Golchin and Mihai Surdeanu. Time travel in LLMs: Tracing data contamination in large language models. In *International Conference on Learning Representations*, 2024.

Carlos Gómez-Rodríguez and Paul Williams. A confederacy of models: A comprehensive evaluation of LLMs on creative writing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14504–14528, 2023.

Google. Gemini 3.6 Flash. Gemini API model documentation, 2026. URL https://ai.google.dev/gemini-api/docs/models/gemini-3.6-flash.

Google DeepMind. Gemini 3.1 Pro Model Card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, February 2026.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Tarun Gupta and Danish Pruthi. All that glitters is not novel: Plagiarism in AI generated research. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25721–25738, 2025.

Zhaoyi Joey Hou, Bowei Alvin Zhang, Yining Lu, Bhiman Kumar Baghel, Anneliese Brei, Ximing Lu, Meng Jiang, Faeze Brahman, Snigdha Chaturvedi, Haw-Shiuan Chang, Daniel Khashabi, and Xiang Lorraine Li. CreativityPrism: A cross-domain evaluation framework for large language model creativity. *Transactions on Machine Learning Research*, 2026.

Xiang Hu, Hongyu Fu, Jinge Wang, Yifeng Wang, Zhikun Li, Renjun Xu, Yu Lu, Yaochu Jin, Lili Pan, and Zhenzhong Lan. Nova: An iterative planning and search approach to enhance novelty and diversity of LLM generated ideas. *arXiv preprint arXiv:2410.14255*, 2024.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Tianfu Wang, Zhengyu Chen, Nicholas Jing Yuan, Jianxun Lian, Kaize Ding, and Hui Xiong. Explaining length bias in LLM-based preference evaluations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6763–6794. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.358. URL https://aclanthology.org/2025.findings-emnlp.358/.

Mete Ismayilzada, Antonio Laverghetta Jr., Simone A. Luchini, Reet Patel, Antoine Bosselut, Lonneke van der Plas, and Roger E. Beaty. Creative preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 9580–9609, 2025. doi: 10.18653/v1/2025.findings-emnlp.509.

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.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, Alon Albalak, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, et al. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling (COLM)*, 2025.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, 2016.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, Yu Rong, Deli Zhao, Tian Feng, and Lidong Bing. Chain of ideas: Revolutionizing research via novel idea development with LLM agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8971–9004, 2025. doi: 10.18653/v1/2025.findings-emnlp.477.

Ruochen Li, Liqiang Jing, Chi Han, Jiawei Zhou, and Xinya Du. LDC: Learning to generate research idea with dynamic control. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, 2023. doi: 10.18653/v1/2023.acl-long.687.

Jianxing Liao, Tian Zhang, Xiao Feng, Yusong Zhang, Haorui Wang, Bosi Wen, Ziying Wang, and Runzhi Shi. RLMR: Reinforcement learning with mixed rewards for creative writing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. NOVER: Incentive training for language models via verifier-free reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7439–7458, 2025. doi: 10.18653/v1/2025.emnlp-main.378.

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.

Li-Chun Lu, Miri Liu, Pin-Chun Lu, Yufei Tian, Shao-Hua Sun, and Nanyun Peng. Rethinking creativity evaluation: A critical analysis of existing creativity evaluations. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6329–6352, 2026. doi: 10.18653/v1/2026.eacl-long.297.

Ximing Lu, Melanie Sclar, Skyler Hallinan, Niloofar Mireshghallah, Jiacheng Liu, Seungju Han, Allyson Ettinger, Liwei Jiang, Khyathi Chandu, Nouha Dziri, and Yejin Choi. AI as humanity's salieri: Quantifying linguistic creativity of language models via systematic attribution of machine text against web text. In *International Conference on Learning Representations*, 2025.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023. doi: 10.1162/tacl_a_00536.

Mistral AI. Mistral-Large-Instruct-2411 model card. Hugging Face, 2024. URL https://huggingface.co/mistralai/Mistral-Large-Instruct-2411.

Mistral AI. Mistral-Small-24B-Instruct-2501 model card. Hugging Face, 2025. URL https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501.

Ludovico Mitchener, Angela Yiu, Benjamin Chang, Mathieu Bourdenx, Tyler Nadolski, et al. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Jay A. Olson, Johnny Nahas, Denis Chmoulevitch, Simon J. Cropper, and Margaret E. Webb. Naming unrelated words predicts creativity. *Proceedings of the National Academy of Sciences*, 118 (25):e2022340118, 2021. doi: 10.1073/pnas.2022340118.

OpenAI. GPT-4 Turbo. OpenAI API model documentation, snapshot gpt-4-turbo-2024-04-09, 2024a. URL https://developers.openai.com/api/docs/models/gpt-4-turbo.

OpenAI. text-embedding-3-small. OpenAI API model documentation, 2024b. URL https://developers.openai.com/api/docs/models/text-embedding-3-small.

OpenAI. GPT-5.2. OpenAI API model documentation, 2025a. URL https://developers.openai.com/api/docs/models/gpt-5.2.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025b. doi: 10.48550/arXiv.2508.10925. URL https://arxiv.org/abs/2508.10925.

OpenAI. GPT-5.5 System Card. https://deploymentsafety.openai.com/gpt-5-5, April 2026a.

OpenAI. GPT-5.6 Sol. OpenAI API model documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, et al. GPT-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

Richard Yuanzhe Pang, Vishakh Padmakumar, Thibault Sellam, Ankur P. Parikh, and He He. Reward gaming in conditional text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 4746–4763. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.262. URL https://aclanthology.org/2023.acl-long.262/.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Kyeongman Park, Nakyeong Yang, and Kyomin Jung. Avoidance decoding for diverse multi-branch story generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7489–7505, 2025. doi: 10.18653/v1/2025.emnlp-main.381.

Kyeongman Park, Minha Jhang, and Kyomin Jung. A universal avoidance method for diverse multi-branch generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15857–15870, 2026.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *International Conference on Learning Representations*, 2018.

Qwen Team. Qwen2.5-72B-Instruct model card. Hugging Face, 2024. URL https://huggingface.co/Qwen/Qwen2.5-72B-Instruct.

Qwen Team. Qwen3.5-122B-A10B-FP8 model card. Hugging Face, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-122B-A10B-FP8.

Qwen Team. Qwen3.5-9B model card. Hugging Face, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-9B.

Qwen Team. Qwen3.6-27B model card. Hugging Face, 2026c. URL https://huggingface.co/Qwen/Qwen3.6-27B.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Mark A. Runco and Garrett J. Jaeger. The standard definition of creativity. *Creativity Research Journal*, 24(1):92–96, 2012. doi: 10.1080/10400419.2012.650092.

Arkadiy Saakyan, Najoung Kim, Smaranda Muresan, and Tuhin Chakrabarty. Death of the novel(ty): Beyond n-gram novelty as a metric for textual creativity. In *International Conference on Learning Representations*, 2026.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10776–10787, 2023.

Moein Salimi, Babak Hosseini Mohtasham, Amin Aghakasiri, Mahdi Naieni, Amir Hossein Qeysarbeigi, Mohammad Masih Shalchian Nazer, Zahra Azar, Mahdi Jafari Siavoshani, and Mohammad Hossein Rohban. Debate as reward: A multi-agent reward system for scientific ideation via RL post-training. *arXiv preprint arXiv:2604.16723*, 2026.

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.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations*, 2025.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. The ideation–execution gap: Execution outcomes of LLM-generated versus human research ideas. In *International Conference on Learning Representations*, 2026.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724*, 2024. doi: 10.48550/arXiv.2405.01724. URL https://arxiv.org/abs/2405.01724.

Min Sen Tan, Zachary Kit Chun Choy, Syed Ali Redha Alsagoff, Nadya Yuki Wangsajaya, Mohor Banerjee, Swaagat Bikash Saikia, and Alvin Chan. Automated creativity evaluation of language models across open-ended tasks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 23139–23173, 2026. doi: 10.18653/v1/2026.acl-long.1061.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024. doi: 10.18653/v1/2024.acl-long.511.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-preference bias in LLM-as-a-judge. In *NeurIPS 2024 Workshop on Safe Generative AI*, 2024.

Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. CycleResearcher: Improving automated research via automated review. In *International Conference on Learning Representations*, 2025.

Yuhao Wu, Yushi Bai, Zhiqiang Hu, Roy Ka-Wei Lee, and Juanzi Li. LongWriter-Zero: Mastering ultra-long text generation via reinforcement learning. In *International Conference on Learning Representations*, 2026.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, Maosong Sun, and Tat-Seng Chua. RLPR: Extrapolating RLVR to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025b.

Z.ai. GLM-4.5-Air model card. Hugging Face, 2025. URL https://huggingface.co/zai-org/GLM-4.5-Air.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating language models for humanlike diversity. In *Conference on Language Modeling*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, 2023.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. In *International Conference on Learning Representations*, 2026.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

A Appendix

A.1 Derivations for the novelty score

A.1.1 Raw surprisal decomposition

Fix q and assume $p_V(x \mid q) > 0$ whenever $\pi_\theta(x \mid q) > 0$. Adding and subtracting the policy log-probability in Equation 2 gives

$$\mathbb{E}_{x\sim\pi_\theta(\cdot|q)}[-\log p_V(x \mid q)] = -\sum_x \pi_\theta(x \mid q)\log \pi_\theta(x \mid q) + \sum_x \pi_\theta(x \mid q)\log \frac{\pi_\theta(x \mid q)}{p_V(x \mid q)} \quad (11)$$

$$= H(\pi_\theta(\cdot \mid q)) + D_{\mathrm{KL}}(\pi_\theta(\cdot \mid q) \,\|\, p_V(\cdot \mid q))\,. \quad (12)$$

This identity holds for the full, unnormalized sequence log-probability.

A.1.2 IDEALIZED INTERPRETATION AFTER PARAPHRASE CORRECTION

Fix a context c, let $b \sim \pi_\theta(\cdot \mid c)$ be a sentence, and let $z = P(b)$ be its deterministic paraphrase, which induces the pushforward distribution $\pi_P(z \mid c) = \sum_{b:P(b)=z} \pi_\theta(b \mid c)$. With full-sequence log-probabilities and the exact baseline $\mu_V^{\mathrm{seq}}(c) = \mathbb{E}_{y\sim p_V(\cdot|c)}[\log p_V(y \mid c)] = -H(p_V(\cdot \mid c))$, the idealized corrected score $S_P^{\mathrm{seq}}(b \mid c) = \mu_V^{\mathrm{seq}}(c) - \log p_V(P(b) \mid c)$ satisfies

$$\mathbb{E}_{b\sim\pi_\theta(\cdot|c)}[S_P^{\mathrm{seq}}(b \mid c)] = -H(p_V(\cdot \mid c)) + H(\pi_P(\cdot \mid c)) + D_{\mathrm{KL}}(\pi_P(\cdot \mid c) \,\|\, p_V(\cdot \mid c)) \,. \tag{13}$$

Thus, paraphrase correction retains the entropy–KL decomposition, now measured after surface realizations are merged by P, up to the constant $-H(p_V)$. Equation 6 approximates this construction with mean token log-probabilities, sentence-level segmentation, and a finite-sample baseline, so the identity motivates rather than exactly describes the implemented score.

A.2 PARAPHRASE-CORRECTION IMPLEMENTATION

We split an abstract at sentence-final punctuation and retain spans of at least 25 characters. V rewrites each sentence independently with greedy decoding and at most 160 new tokens, seeing only the sentence and the following instruction:

> Rewrite the given sentence from an ML paper abstract in your own words. Preserve every technical claim, method name, and number exactly; change the wording and sentence structure. Keep plain technical prose. Output only the rewritten sentence.

The rewrite is scored by teacher forcing in the original context c_i, whose preceding sentences are not rewritten. The procedure is deterministic for a fixed checkpoint and sentence split.

A.3 PARAPHRASE FIDELITY

We check that paraphrasing does not change what a sentence claims on the 1,035 sentences of the 130 human-written abstracts, where the rewrite has the most stylistic compression to undo.

Table 5: Cosine distance (text-embedding-3-small) between a sentence and its paraphrase, compared with the distance to a different sentence of the same abstract and to the sentence at the same position of another sample for the same prompt (sibling).

	Sentences	Sentence vs. paraphrase	Other sentence	Sibling sample
Human abstracts	1,035	0.0778	0.5405	—
Base	4,448	0.0726	0.5192	0.3446
DPO_{CD}	4,534	0.0758	0.5207	0.3303
GRPO_{CD}	4,722	0.0800	0.5320	0.3722

Table 6: GPT-5.6 Sol verdicts on whether a sentence and its paraphrase, shown in random order, state the same proposition (1,035 sentences from human-written abstracts). *Predictability gain* is the change in log-probability under V from scoring the paraphrase instead of the sentence.

Verdict	Share	Sentences	Predictability gain (median)
SAME, no claim differs	74.4%	770	-0.132
MINOR, a claim is weakened or made vaguer	12.9%	133	-0.071
DIFFERENT, a claim is added or lost	12.8%	132	-0.093

As shown in Table 5, a paraphrase is four to seven times closer to its source sentence than to any other sentence, and fidelity on human abstracts is within 0.008 of that on model outputs. In Table 6, under a strict rubric in which any added or dropped qualifier counts, 87% of rewrites differ by no more than hedging, and the DIFFERENT cases mostly involve discourse framing or near-synonyms (e.g., "off-the-shelf" $\rightarrow$ "pre-trained"). Crucially, meaning drift does not increase predictability: the median gain is no larger for DIFFERENT than for SAME, and the correlation between embedding drift and predictability gain is negative for every model output set (-0.26 to -0.28, $p < 10^{-70}$) and for human abstracts (-0.072).

A.4 JUDGES USED IN THE EVALUATOR DIAGNOSTICS

The LLM judges row of Table 1 combines different judges per column. OS is the mean of GPT-5.2, GPT-5.6 Sol, and Gemini 3.6 Flash on the DPO$_{CD}$/Base pairs of Section 3.3.3. LP is the mean of seven judges on 1,189 published–model pairs: Mistral-Large-Instruct-2411, GPT-4 Turbo (OpenAI, 2024a), GPT-4o (OpenAI et al., 2024), Llama-3.1-70B-Instruct, Qwen2.5-72B-Instruct (Qwen Team, 2024), GPT-5.5 (OpenAI, 2026a), and Gemini 3.1 Pro (Google DeepMind, 2026). The LP value of Constructive Divergence is computed on the same pairs (Appendix B.9). CP is GPT-5.2. TR is Claude Opus 5 (Anthropic, 2026), the highest of 23 judges tested. RΔ is the mean of GPT-5.5 and Gemini 3.1 Pro. Human refers to the outputs selected by GPT-5.2.

A.5 CONSTRUCTION OF EVALUATOR-DISAGREEMENT PAIRS

We start from 418 abstracts for 119 held-apart keyword sets, disjoint from both the training and evaluation prompts (338 model-generated and 80 published ICLR 2025 abstracts), and pair abstracts only within the same keyword set. After the safeguards in Appendix A.10, each abstract receives $\mathrm{C}(x, q) = S(x \mid q)/0.37 + R(x \mid q)/0.90$, and Constructive Divergence selects the higher-C abstract. GPT-5.2 sees both abstracts and the keywords and is asked which presents the more original idea, ignoring wording, length, and writing quality. We query it three times in each presentation order and retain a pair only when both order-wise majorities select the same abstract and that abstract is the lower-C one.

We sample at most one pair per keyword set, yielding 16 model–model and 10 published–model pairs. The absolute length difference is capped at 130 (model–model) and 160 (published–model) displayed characters, with realized medians of 62.5 and 83. Within each stratum, the higher-C abstract is the longer one in exactly half of the pairs, and its left/right position is balanced. The interface hides sources, evaluators, and scores. The study contains these 26 pairs plus four position-reversed repeats for rater self-consistency; the Human column of Table 1 uses only the 26 unique pairs.

A.6 HUMAN EVALUATION DETAILS

All human studies are rated by three PhD-level raters. Table 7 reports each rater's decisions on the unique pairs, excluding position-swapped repeats, and Table 8 reports pairwise agreement between raters on pairs where both express a preference.

Table 7: Per-rater decisions. For the model comparisons, wins are for the trained policy; for the disagreement pairs, wins are for the higher-C output against the output selected by GPT-5.2. p is a two-sided sign test excluding ties.

Study	Rater	Win : Loss	Ties	Win rate	p
GRPO$_{CD}$ / Base	Rater 1	7 : 17	3	29.2%	0.06
	Rater 2	8 : 7	12	53.3%	1.00
	Rater 3	11 : 12	4	47.8%	1.00
	Majority	10 : 15	2	40.0%	0.42
DPO$_{CD}$ / Base	Rater 1	12 : 13	2	48.0%	1.00
	Rater 2	5 : 7	15	41.7%	0.77
	Rater 3	15 : 10	2	60.0%	0.42
	Majority	12 : 10	5	54.5%	0.83
CD / GPT-5.2	Rater 1	15 : 10	1	60.0%	0.42
	Rater 2	4 : 8	14	33.3%	0.39
	Rater 3	15 : 11	0	57.7%	0.56
	Majority	11 : 9	6	55.0%	0.82

As shown in Table 7, no individual rater shows a significant preference in any study, and the direction differs across raters: Rater 3 prefers DPO$_{CD}$, whereas Raters 1 and 2 slightly prefer the base policy. Rater 2 marks nearly half of the pairs as ties. Pairwise agreement ranges from 36% to 86% (Table 8), close to the 50% expected by chance for most rater pairs.

Table 8: Pairwise agreement between raters on pairs where both raters express a preference.

Study	Raters 1–2	Raters 1–3	Raters 2–3
GRPO_{CD} / Base	12/14 (86%)	14/21 (67%)	5/12 (42%)
DPO_{CD} / Base	6/11 (55%)	14/23 (61%)	7/12 (58%)
CD / GPT-5.2	4/11 (36%)	17/25 (68%)	5/12 (42%)

A.7 Choosing the reference model

Table 9: Reference-model selection probes, all scored with the sentence-level estimator of the main experiments. P1: AUC for ranking post-cutoff abstracts (arXiv cs.LG, March–August 2026, after the training cutoff of every candidate) above pre-cutoff ones (July 2022–June 2023; 0.5 is no signal). P2: AUC for ranking classic ideas above post-cutoff ones (lower is better). P3: median absolute score change under paraphrase divided by the across-idea interquartile range (lower is better). P4: fraction of abstracts whose score rises when their sentences are shuffled (lower is better). Candidates are gpt-oss-120B (OpenAI, 2025b), Llama-3.1-70B-Instruct (Grattafiori et al., 2024), Qwen3.5-122B-A10B (Qwen Team, 2026a), and Mistral-Large-Instruct-2411 (Mistral AI, 2024).

Candidate	P1 cutoff era ↑	P2 classic ideas ↓	P3 surface ↓	P4 shuffle leak ↓
gpt-oss-120B	**0.743**	0.294	0.40	**16%**
Llama-3.1-70B	0.557	0.149	0.42	59%
Qwen3.5-122B	0.558	**0.084**	0.35	66%
Mistral-Large-123B	0.481	0.146	**0.32**	70%

We treat P1 as decisive because it directly asks whether the model finds genuinely new work surprising. As shown in Table 9, only gpt-oss-120B separates post-cutoff from pre-cutoff abstracts (0.743), while the others remain at 0.48–0.56. P4 agrees: shuffling sentences destroys the argument but keeps the vocabulary, and only gpt-oss-120B scores the shuffled version lower for most texts. P2 and P3 reward scores that move little, which a model with no signal also satisfies, so we use them only as tiebreakers among candidates that pass P1; gpt-oss-120B is still mid-pack on P3. The probes use 452 candidate texts and the 80 held-apart human-written abstracts of Appendix A.10. Robustness to other reference models is tested separately in Appendix B.2.

A.8 Baselines

Decoding. All policies and baselines use temperature 0.9, top-$p = 0.95$, at most 340 generated tokens, and five samples per test prompt with a fixed seed. Only the $\text{Base}_{t=\ldots}$ baselines change the temperature.

Base prompt. All policies receive the following instruction with the held-out keywords substituted:

```
You are a researcher writing the abstract for an article
submitted to PNAS (Proceedings of the National Academy of
Sciences).
Write the abstract for a PNAS research article whose author
keywords are: {keywords}.
Requirements:
- One paragraph, ~150-220 words.
- Self-contained: motivation, approach, key result,
significance.
- Do not list the keywords; weave the science in naturally.
Output only the abstract text.
```

Creativity Instruction. We insert the following paragraph after the keyword line of the base prompt and keep everything else unchanged:

```
The contribution must be genuinely ORIGINAL: propose an idea
that does not already exist anywhere in the literature as
of 2026.  Do not restate well-known, classic, or textbook
ideas, however important they are; an expert who knows
the field must not be able to name an existing paper that
already did this.
```

Self-Refinement. The base policy revises its own held-out output once, with critique and rewriting in a single call:

```
Below is a draft abstract for a PNAS research article whose
author keywords are:  {keywords}.
Draft:
{draft}
First, silently identify what is conventional or predictable
about the draft's core idea.  Then rewrite the abstract so
that its central idea is substantially more original and
less predictable to an expert in the field, while staying
on the same keywords and keeping the same format (one
paragraph, ~150-220 words; motivation, approach, key result,
significance; do not list the keywords).
Output only the revised abstract text.
```

Best-of-N. We draw 16 base-policy samples per test prompt and select the one with the highest Constructive Divergence; $N \in \{2, 4, 8, 16\}$ are nested prefixes of the same pool. This is the only baseline that uses our score at inference time.

DPO$_{LLM}$. This baseline changes only the ranking function of our DPO recipe (Appendix A.9): GPT-5.2 ranks all eight candidates at once, twice in different random orders, and the two rankings are averaged. Its prompt matches the question used for pairwise judging in Section 3.3.3:

```
Below are {n} abstracts (labeled {labels}) written for the
SAME research keywords:  {keywords}.
Imagine all of them were submitted TODAY (2026) to a top ML
venue and you are an expert reviewer who knows the field
thoroughly.  Rank them from the MOST ORIGINAL idea to
the LEAST -- the one you would be least likely to reject
as 'this has already been done' comes first.  An idea
that already exists in the literature, including famous
foundational work, is not original.  Judge the idea itself
-- not the wording, not the length, not the writing quality.
Reply with the {n} labels in ranked order, separated by
commas, and nothing else.
{abstracts}
Ranking:
```

A.9 Training details

DPO. For each of the 1,000 training prompts, we sample eight candidates from the base policy at temperature 1.0 and top-$p = 0.95$, standardize surprisal and recoverability within the prompt, and rank candidates by $z_q(S) + z_q(R)$. Each of the two highest-ranked candidates that pass the safeguards is paired with each of the two lowest-ranked candidates when their score gap is at least 0.5, producing 4,000 preference pairs. We train for three epochs with $\beta = 0.1$, a learning rate of 10^{-5}, a batch size of 2, and four gradient-accumulation steps.

GRPO. We use GRPO with the token-level loss normalization of DAPO (Yu et al., 2025a), not the full DAPO algorithm. Each optimizer step uses four prompts with eight completions each (temperature 1.0, top-$p = 0.95$, at most 340 tokens), i.e., 32 completions. We train a rank-32 LoRA adapter for 600 steps with a per-device batch size of 8, four gradient-accumulation steps, a KL coefficient of 0.02, and a learning rate of 10^{-5} with 30 warmup steps and a cosine schedule floored at 0.2 of

the peak. The reward of each completion is $\mathrm{C}(x, q)$ minus the safeguard penalties, and group-wise advantages follow standard GRPO. Each safeguard violation subtracts 5 from the reward.

A.10 SAFEGUARDS

Neither term penalizes an unreadable output by itself: rare-token strings surprise V, and verbatim keyword lists are easy for D to invert. Two filters therefore act as constraints outside the score.

Perplexity ceiling. We score 80 held-apart human-written ICLR 2025 abstracts, disjoint from both the training and evaluation prompts, under D as for model outputs and set the ceiling at their 95th percentile, 21.1 (median 11.9, interquartile range 9.3–15.4).

Degeneration check. An output is rejected if it has fewer than 40 or more than 400 words, if any 4-gram occurs three or more times, or if its distinct-to-total whitespace-token ratio is below 0.45.

Application. GRPO subtracts a penalty of 5 per violation, which exceeds the within-group score spread. DPO bars violating candidates from the preferred side of a pair, while the rejected side is drawn from the bottom of the ranking regardless. Neither threshold was tuned. Among the 650 held-out outputs of each of Base, GRPO$_{CD}$, DPO$_{CD}$, Creativity Instruction, Self-Refinement, and DPO$_{LLM}$, no output violates either filter; only Base$_{t=1.8}$ exceeds the ceiling (200 of 650 outputs). Because the filters never fire on the trained policies, these experiments do not test recovery from active reward hacking.

B ADDITIONAL ANALYSIS

B.1 PROMPT-PAIRED OPTIMIZATION STATISTICS

For each method, we average the five samples per prompt and compute the method-minus-base difference Δ_q over the 130 held-out prompts. Table 10 reports the mean difference and the paired standardized mean difference $d = \mathrm{mean}(\Delta_q)/\,\mathrm{sd}(\Delta_q)$; Best-of-$N$ compares its selected output with the five-sample base mean. C rises on 129 of the 130 prompts for DPO$_{CD}$ and on all 130 for GRPO$_{CD}$.

Table 10: Prompt-paired changes against the base policy, with paired standardized mean differences in parentheses.

Method	ΔS (d)	ΔR (d)	ΔC (d)
GRPO$_{CD}$	$+0.591$ (2.70)	$+0.161$ (0.64)	$+1.775$ (2.70)
DPO$_{CD}$	$+0.453$ (1.90)	$+0.374$ (1.19)	$+1.639$ (2.09)
Creativity Instruction	$+0.029$ (0.12)	-0.392 (-0.89)	-0.357 (-0.45)
Self-Refinement	$+0.176$ (0.65)	-0.370 (-0.92)	$+0.064$ (0.07)
DPO$_{LLM}$	-0.028 (-0.13)	-0.092 (-0.34)	-0.178 (-0.26)
Best-of-2	$+0.171$ (0.58)	$+0.015$ (0.04)	$+0.480$ (0.57)
Best-of-4	$+0.263$ (0.97)	$+0.107$ (0.30)	$+0.829$ (1.05)
Best-of-8	$+0.412$ (1.39)	$+0.166$ (0.47)	$+1.297$ (1.73)
Best-of-16	$+0.515$ (1.98)	$+0.143$ (0.42)	$+1.550$ (2.29)

B.2 ROBUSTNESS TO REFERENCE-MODEL CHOICE

We re-score with Qwen3.5-9B (Qwen Team, 2026b), Qwen3.6-27B (Qwen Team, 2026c), Qwen2.5-72B-Instruct (Qwen Team, 2024), Qwen3.5-122B-A10B-FP8 (Qwen Team, 2026a), GLM-4.5-Air (Z.ai, 2025), Mistral-Small-24B-Instruct-2501 (Mistral AI, 2025), Mistral-Large-Instruct-2411 (Mistral AI, 2024), and Llama-3.1 (Grattafiori et al., 2024). As shown in Table 11, all seven alternative Vs, spanning 9B to 122B parameters and four families, reproduce the surprisal gain ($d = 0.58$–1.36), all seven alternative Ds preserve the recoverability gain, and DPO$_{CD}$ exceeds GRPO$_{CD}$ under every model. Agreement on which individual output scores higher is 70% across reference models, so robustness holds at the level of policy effects rather than individual pairwise verdicts.

Table 11: Paired effect sizes against the base policy when the same held-out outputs are re-scored with reference models not used in training.

Reference V	DPO$_{CD}$	GRPO$_{CD}$	Second reference D	DPO$_{CD}$	GRPO$_{CD}$
Qwen3.5-9B	1.15	0.94	Llama-3.1-8B	1.22	0.28
Qwen3.6-27B	1.15	0.89	Mistral-Small-24B	1.16	0.28
Llama-3.1-70B	0.58	0.52	Qwen3.6-27B	1.06	0.33
Qwen2.5-72B	0.99	0.74	Qwen2.5-72B	1.18	0.27
GLM-4.5-Air	0.74	0.70	GLM-4.5-Air	1.07	0.21
Qwen3.5-122B	1.21	0.79	Qwen3.5-122B	1.13	0.27
Mistral-Large	1.36	0.75	Mistral-Large	0.91	0.22

B.3 Temperature versus training

Raising the temperature trades recoverability for surprisal. At $t = 1.5$, Base gains 0.134 in S, loses 0.022 in R, and its perplexity rises from 6.33 to 8.07; DPO$_{CD}$ gains 3.4 times as much surprisal while also raising R at lower perplexity. At $t = 1.8$, perplexity rises to 18.83. The paraphrase correction is what exposes this difference: without it, the context-calibrated surprisal of Base$_{t=1.8}$ rises by $+1.08$, but only by $+0.186$ after correction, so nearly 90% of the apparent gain is wording, whereas the trained policies lose far less.

Under teacher forcing on the same texts, training sharpens the policy rather than randomizing it: token-level entropy falls from 0.473 to 0.218, whereas $t = 1.8$ raises it to 0.829. The trained policy diverges from the base by 0.53 nats per token, 84% of it on content words.

B.4 Additional idea-shift and diversity statistics

Table 12: Prompt-paired changes in the idea coding of Table 3(a), with paired standardized mean differences in parentheses. † marks changes that are not significant.

	Recombination	Δ (d)	Specificity	Δ (d)
Base	0.529	—	3.31	—
GRPO$_{CD}$	0.480	-0.049 (0.24)	3.11	-0.202 (0.59)
DPO$_{CD}$	0.448	-0.082 (0.37)	3.33	$+0.019^{\dagger}$ (0.06)
Base$_{t=1.5}$	0.505	$-0.025^{\dagger}$ (0.12)	3.41	$+0.100$ (0.28)
DPO$_{LLM}$	0.600	$+0.071$ (0.27)	3.86	$+0.552$ (1.18)

Idea coding. The coder sees one abstract at a time and never sees a score. As shown in Table 12, recombination under DPO$_{CD}$ rises on only 12 of 130 prompts, whereas raising the temperature leaves it essentially unchanged. GRPO$_{CD}$ lowers specificity on a paired basis ($d = 0.59$), while DPO$_{LLM}$ is both the most recombinative and the most specific ($d = 1.18$), i.e., it sounds more particular while being more derivative. Gemini 3.6 Flash (temperature 0) receives each abstract with the following instruction; recombination and specificity in Table 3(a) are the `is_combination` and `specificity` fields, idea types are the `novelty_type` field, and the extracted core ideas of Appendix B.5 are the `core_idea` field.

```
You extract the core research idea from a machine-learning
paper abstract.  Ignore results, numbers and promotional
claims.  Return ONLY a JSON object with keys:
core_idea:  one sentence (<=30 words) stating the central
idea as a specific mechanism or claim, not a topic;
problem:  the problem or gap addressed (<=20 words);
mechanism:  how the idea works, the key technical move
(<=25 words);
novelty_type:  exactly one of [new_architecture_or_model,
new_objective_or_training_signal,
new_algorithm_or_inference_procedure, new_theory_or_analysis,
new_problem_or_task_formulation, new_benchmark_or_dataset,
```

```
application_or_domain_transfer,
combination_of_existing_methods, other];
builds_on: list of up to 3 named existing methods/concepts
the idea builds on (short names);
is_combination: true if the idea is mainly a combination of
existing components, else false;
specificity: integer 1-5, 1 = vague/generic (could describe
many papers), 5 = concrete and distinctive;
buzzwords: list of generic hype words present (e.g. novel,
scalable, paradigm, unified, framework), possibly empty.
```

Where the surprisal score peaks. The concentration in Table 3(b) is the share of maximum-S_i sentences with a role divided by that role's share of all sentences, where $S_i = \widehat{\mu}_V(c_i) - \ell_V(\tilde{s}_i \mid c_i)$. The maximum-$S_i$ sentence rarely coincides with the extracted idea sentence (6.8% for DPO$_{CD}$ against a 14.4% chance rate; within-abstract rank correlation -0.28). Roughly one third of maximum-S_i sentences are result statements, which are unverified in a generated abstract, so the score cannot yet distinguish a surprising claim from an invented one. The rhetorical roles are assigned by Gemini 3.6 Flash (temperature 0), which receives the numbered sentences of each abstract with the following instruction; Table 3(b) reports the MECHANISM_DETAIL, RESULT, CONTRIBUTION_STATEMENT, and IMPLICATION roles.

```
You label the rhetorical move each sentence of a
machine-learning paper abstract makes. You are given the
sentences numbered in order. Label every sentence with
exactly one of:
BACKGROUND: states what the field does or why the area
matters, no specific gap;
PROBLEM: states the gap, limitation or failure the paper
addresses;
CONTRIBUTION_STATEMENT: announces what the paper proposes or
introduces;
MECHANISM_DETAIL: explains how the proposed thing works, its
components or technical move;
SETUP_OR_SCOPE: describes datasets, benchmarks, tasks or
experimental setting;
RESULT: reports an outcome, number or comparison;
IMPLICATION: states broader significance, future work or
what the work suggests.
Return ONLY a JSON object {"labels": ["<label for sentence
1>", "<label for sentence 2>", ...]} with exactly one label
per sentence, in order.
```

Table 13: Embedding geometry for all policies and baselines, extending Table 4. *Ratio* is Inner-Diversity relative to Base. Distances use all 2,852 accepted ICLR 2025 papers (2,392 posters, 295 spotlights, and 165 orals), collected through the OpenReview API. † marks changes from Base that are not significant ($p \geq 0.05$, Wilcoxon, paired by prompt).

	Inner-Diversity	Ratio	Human-Distance
Base	0.0793	1.00	0.2905
DPO$_{CD}$	0.0684	0.86	0.2968
GRPO$_{CD}$	0.0805	1.02†	0.2972
DPO$_{LLM}$	0.0752	0.95	0.2851
Creativity Instruction	0.1432	1.81	0.3194
Self-Refinement	0.1265	1.60	0.3074
Base$_{t=1.5}$	0.1016	1.28	0.2907†
Base$_{t=1.8}$	0.1538	1.94	0.3025
DPO$_{CD,\,t=1.5}$	0.0809	1.02	0.2993

Diversity. DPO$_{CD}$ narrows within-prompt diversity by 14% ($d = -0.60$) and reduces distinct idea types per prompt from 1.87 to 1.68, but the extracted core ideas narrow by only 3% ($d = -0.17$), and

cross-prompt diversity is unaffected. DPO_{LLM} also narrows it (5%, $d = -0.22$), so the narrowing is a property of offline preference optimization rather than of our reward. As shown in Table 13, high diversity alone is not the goal: Creativity Instruction and Self-Refinement are the most diverse and furthest from existing abstracts, but they lose more than 0.38 in recoverability and raise perplexity. Both CD-trained policies move away from the nearest real abstract ($d = 0.38$ and 0.47), whereas DPO_{LLM} moves closer ($d = -0.30$, $p = 0.001$). Sampling DPO_{CD} at $t = 1.5$ restores diversity above Base, keeps 101% of its C gain, and further increases Human-Distance ($d = 0.53$), so the diversity cost can be recovered at inference time without retraining.

B.5 Qualitative analysis

We inspect five prompts for which combination-based ideas are common under the base policy and less common under DPO. Within each policy, we select the candidate nearest that policy's median Constructive Divergence score, rather than the most extreme candidate, and Gemini 3.6 Flash extracts the core idea from each abstract without seeing its source policy or score (Table 14); the full abstracts are provided in the supplementary material.

As shown in Table 14, the base output typically names two or three familiar components and proposes combining them, whereas the DPO output states a single relationship or invariant that explains what the mechanism is meant to do. This is consistent with the corpus-level decrease in recombination. Four of the five DPO examples improve both surprisal and recoverability; the preference-learning example raises surprisal but slightly lowers recoverability.

B.6 Full results across all policies and baselines

Published abstracts score below every policy and baseline in Table 15. This is not memorization: asked to name 100 other ICLR 2025 papers, V produces the correct title for none. It is not the paraphrase step either: fidelity is equal on human and model text (Appendix A.3), and without correction human abstracts are more surprising than Base outputs (3.029 vs. 2.888). Most of the gap comes from surprisal; the smaller recoverability part arises because published abstracts do not reuse the author keywords verbatim, whereas prompted policies do (0.84, rising to 0.94 after DPO_{CD}). The remainder suggests that, given its own keywords and after wording is normalized, what a published abstract claims is close to what a strong model expects. This is plausible given how the prompts are built: the author keywords were written to summarize the very paper whose abstract is scored, so conditioning on them already tells V much of what that abstract claims. A policy receives the same keywords without the paper behind them and must commit to a concrete problem and mechanism that the keywords leave open, which is the kind of departure that surprisal rewards. The reversal under paraphrase correction points the same way: published abstracts are more surprising than Base outputs in wording (3.029 vs. 2.888 before correction) but less surprising in content, so their distinctiveness relative to their own keywords lies more in phrasing than in the claims themselves.

Table 14: Selected before–after examples. Each row uses the same keyword prompt and a score-median candidate from each policy.

Prompt topic	Base policy: extracted core idea	DPO policy: extracted core idea
3D editing and diffusion	Combines a multi-view-consistent diffusion backbone with geometry-aware attention in the diffusion latent space.	Injects geometry-conditioned attention into 3D latent diffusion so local edits preserve global topology.
Diffusion, privacy, and mode collapse	Adds dynamic noise scheduling and a diversity regularizer to a differentially private latent diffusion model.	Applies geometry-aware stochastic perturbations during denoising to separate global manifold structure from local memorization.
Causal video VAE and FiLM	Uses a FiLM-inspired conditional latent flow to evolve frame latents causally.	Factorizes static spatial features from temporal motion with an invariance-learning module inside a causal VAE.
Spurious correlation and fairness	Combines distributionally robust optimization with adversarial debiasing under subpopulation shift.	Uses counterfactual invariance to identify and remove non-causal proxy variables while retaining robust predictive signals.
Preference and rule learning	Adds an explicit rule-learning module to a latent choice model.	Recasts preference learning as dynamic rule updating in which sequential rewards induce structured heuristic switches.

Table 15: All policies and baselines on the 130 held-out prompts, with the published abstracts of the same papers scored through the same pipeline. w/o Context & Paraphrase additionally drops the context calibration. Model rows average five samples per prompt; the human row has one abstract per prompt.

	Surprisal S	Recoverability R	Score C	PPL ↓	Words
Base (Gemma 4 12B-it)	3.712	−2.645	7.094	6.33	176.7
GRPO$_{CD}$	4.303	−2.484	8.869	7.55	174.1
DPO$_{CD}$	4.165	−2.271	8.734	7.64	177.2
w/o Context & Paraphrase	4.103	−2.247	8.593	8.31	174.3
w/o Paraphrase	4.102	−2.257	8.577	8.43	175.3
w/o Recoverability	4.399	−2.660	8.933	8.12	174.7
w/o Surprisal	3.759	−2.122	7.802	6.77	176.4
Creativity Instruction	3.741	−3.037	6.736	8.96	181.7
Self-Refinement	3.888	−3.015	7.158	11.52	180.2
DPO$_{LLM}$	3.684	−2.737	6.916	7.18	183.0
Best-of-8	4.124	−2.479	8.391	6.60	176.6
Best-of-16	4.227	−2.502	8.645	6.73	176.8
Base$_{t=1.5}$	3.847	−2.667	7.433	8.07	177.1
Base$_{t=1.8}$	3.898	−2.743	7.486	18.83	183.1
Human-written abstracts	2.315	−3.154	2.752	10.56	185.1

We therefore interpret C as a comparison among outputs for the same prompt, not as an absolute measure of creativity, and never use the human row as a target.

B.7 Why sentences rather than tokens

Table 16: Share of function words and punctuation among the single highest-scoring token of each abstract and among the top 5% of tokens, over 650 abstracts per policy. The remainder is content words and numbers.

	Highest-scoring token		Top 5% of tokens	
	Function word	Punctuation	Function word	Punctuation
Base	15.2%	6.2%	16.0%	6.0%
GRPO$_{CD}$	17.1%	3.8%	16.3%	4.9%
DPO$_{CD}$	19.1%	4.0%	17.6%	5.0%
Base$_{t=1.5}$	9.4%	4.9%	12.6%	4.1%

As shown in Table 16, in roughly one in five base-policy abstracts the most surprising token carries no content; the most frequent are “these”, the sentence-final period, a quotation mark, “and”, and “by”. The pattern holds for both trained policies and for the top 5% of tokens. Content words carry more surprisal on average, but the per-token maximum is unstable in a way that a sentence-level aggregate is not.

B.8 LLM-as-a-judge protocol and preference between the trained policies

The three LLM-as-a-judge models, GPT-5.2, GPT-5.6 Sol, and Gemini 3.6 Flash, compare 260 anonymized pairs per comparison in both presentation orders; a pair counts as decided only when both orders agree. Each judge receives the following prompt:

```
Two ML paper abstracts (A and B) were written for the SAME
research keywords: {kw}.
Imagine both were submitted TODAY (2026) to a top ML
venue and you are an expert reviewer who knows the field
thoroughly.  Which one presents the more ORIGINAL idea --
the one you would be LESS likely to reject as 'this has
```

```
already been done'?  An idea that already exists in the
literature, including famous foundational work, is not
original.  Judge the idea itself -- not the wording, not
the length, not the writing quality.
Reply EXACTLY one character:  A or B.
A:
{A}
B:
{B}
Answer:
```

Human raters see the two anonymized abstracts side by side and answer a single question on a five-point scale (A is much more creative, A is somewhat more creative, similar, B is somewhat more creative, B is much more creative).

> Which one is more creative? Here, creative means that the work presents an idea that is new relative to existing research and, at the same time, that it is clear what the research is about. A text that satisfies only one of the two is considered less creative.

Table 17: Judged preference between DPO_{CD} and GRPO_{CD} (260 pairs per judge).

Judge	Prefers DPO_{CD}	Prefers GRPO_{CD}	Ties
GPT-5.2	96	69	95
GPT-5.6 Sol	86	80	94
Gemini 3.6 Flash	103	78	79
Total	285 (55.7%)	227	268

GRPO_{CD} leads DPO_{CD} in C by 0.135, on 73 of 130 prompts, yet the judges prefer DPO_{CD}, the policy with the larger recoverability gain (Table 17). The effect is small: only GPT-5.2 is significant on its own ($p = 0.043$; 0.074 and 0.70 for the others), and a third of pairs are ties. We read this as the judges weighting the two terms differently, not as a ranking of the two methods.

B.9 Shortcut checks

Table 18: Surface statistics of the 650 held-out outputs of each policy.

	Base	DPO_{CD}	GRPO_{CD}
Share of prompt keywords appearing verbatim	0.839	0.941	0.874
Repeated 3-gram rate	0.0008	0.0006	0.0004
Sentences per abstract	6.84	6.98	7.26
Type–token ratio (alphabetic words)	0.726	0.737	0.751
"novel" per abstract	0.88	0.92	0.95
Words per abstract	176.7	177.2	174.1
Distinct 4-grams (pooled over the policy)	0.776	0.742	0.784
Contains "we propose / we introduce"	0.632	0.248	0.454
Contains "we present a novel"	0.140	0.394	0.308
Contains "a scalable pathway"	0.286	0.366	0.231
Contains a "significance" word	0.843	0.859	0.840

Table 18 counts surface shortcuts on the 650 held-out outputs of each policy. Neither policy lengthens its outputs, repeats itself, or inflates self-described novelty ("novel" per abstract $0.88 \to 0.92$ and 0.95). Two changes are real. First, DPO_{CD} raises verbatim keyword inclusion from 0.84 to 0.94 (GRPO_{CD}: 0.87), which accounts for part of its recoverability gain, since D recovers keywords more easily when they appear in the text. Second, the framing of the contribution shifts from "we propose / we introduce" (63% $\to$ 25%) to "we present a novel" (14% $\to$ 39%), and DPO_{CD} slightly concentrates its phrasing (distinct 4-grams $0.776 \to 0.742$), which GRPO_{CD} does not (0.784). The phrase "a scalable pathway", a habit of the base model present in 25% of the training candidates and equally frequent in chosen and rejected DPO samples, is not selected by the reward.

Table 19: Longer-output preference on identical pairs: the share of pairs in which each evaluator ranks the longer abstract higher. The pairs are those judged in the length-bias diagnostic, built from 300 ICLR 2025 papers and four model-generated abstracts per paper; Constructive Divergence scores all 1,500 texts with the same pipeline as the main experiments. p is a two-sided binomial test against 50%.

Pairs	Evaluator	Longer preferred	p
Published–model (1,181–1,189)	Seven LLM judges	55.1–62.4% (mean 58.8%)	all $< 10^{-3}$
	Constructive Divergence	46.7%	0.027
Model–model (1,763)	GPT-4o	55.8%	1.2×10^{-6}
	GPT-5.2	63.2%	1.1×10^{-28}
	Constructive Divergence	52.2%	0.070

Table 19 compares the evaluators on identical pairs. On model–model pairs, where no judge can recognize a published paper, both judges prefer the longer abstract, whereas Constructive Divergence does not differ significantly from 50%. On published–model pairs, Constructive Divergence ranks the model output higher in 90.9% of pairs, consistent with Appendix B.6, and among those pairs it ranks the longer abstract higher in 48.6%. The deviation below 50% comes from the 108 pairs in which the published abstract wins, where the published abstract is the shorter one in 72%. Within the same policy and decoding setting, Constructive Divergence is also length-neutral on 8,655 same-prompt pairs, preferring the longer abstract 50.1% of the time ($p = 0.80$).