When Does Critique Improve AI-Assisted Theoretical Physics? SCALAR: Structured Critic–Actor Loop for Agentic Reasoning

Anonymous Authors[1]

Abstract

As large language models (LLMs) show increasing promise on research-level physics reasoning tasks and agentic AI becomes more common, a practical question emerges: *How does the interaction between researchers and agents affect the results?* We study this using SCALAR (Structured Critic–Actor Loop for Agentic Reasoning), an Actor–Critic–Judge pipeline applied to quantum field theory and string theory problems. The Actor proposes solutions, the Critic provides iterative feedback, and an independent Judge evaluates the transcript against reference solutions. We vary the Actor persona, the Critic feedback strategy, and the Actor model family and scale. Multi-turn dialogue improves over single-shot attempts throughout, but both the mechanism of improvement and the value of different prompting choices depend strongly on the Actor–Critic pairing. Increasing the scale within one model family (*e.g.* from the 8B-parameter DeepSeek-R1 variant to DeepSeek-R1 70B) improves some easier-problem behavior, but does not remove the hardest bottleneck we observe. Critic feedback strategy matters most clearly in the asymmetric Actor–Critic setting (*e.g.*, a lightweight Haiku Actor guided by a stronger Sonnet Critic), where constructive feedback improves mean-score outcomes. In same-family Actor–Critic settings, strategy effects are weaker: lenient feedback is sometimes favored, while strict and adversarial feedback are not beneficial. Taken together, SCALAR provides a controlled testbed for evaluating which interaction structures help or hinder AI-driven scientific discovery.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the AI4Physics Workshop at the International Conference on Machine Learning (AI4Physics@ICML 2026). Do not distribute.

1. Introduction

Large language models (LLMs) and LLM-based agents are a new type of interlocutor in the dialogue that drives the scientific process. They can reason, make decisions (rather than merely perform algorithmic computations), and even exhibit greater adaptability to iterative prompts than one-shot queries. This behavior is closer to that of a human collaborator than that of any previous computational tool. Early evidence of new contributions to theoretical physics (Guevara et al., 2026; Schwartz, 2026; Shih, 2026b;a; Lu et al., 2025), mathematical discovery (Romera-Paredes et al., 2024; Novikov et al., 2025), and agentic scientific workflows in high-energy physics (Plehn et al., 2026; Agrawal et al., 2026) is very encouraging. However, how physicists should structure this collaboration is an open question: in general, multi-turn interactions exhibit sticky error states and capability degradations (Liang et al., 2024; Laban et al., 2025; Zhang et al., 2025b), although structured multi-agent pooling can reduce hallucinations (Till et al., 2025).

We probe these dynamics with SCALAR, a deliberately pedagogical *Actor–Critic–Judge* pipeline: one LLM agent (comparable to a student) plays the *Actor* attempting a graduate-level quantum-field-theory (QFT) or string theory problem. The *Critic* LLM (analogous to a teaching assistant) then delivers formative feedback mid-task, before the final agent playing the *Judge* (teacher) sets the standard against which the work is ultimately evaluated; for prior discussions of LLMs as Judges see e.g. (Zheng et al., 2023). Such a pedagogical interpretation completes a scaffolding in the sense of educactional psychology (Wood et al., 1976; Vygotsky, 1978). Multi-agent approaches to LLM reasoning have shown promise — from debate frameworks (Du et al., 2024; Estornell et al., 2024) to specialized refinement agents for physics (Jaiswal et al., 2024) and interpretable AI-scientist collaboration (Xu et al., 2025). Pre-prompting strategies have also been shown to make measurable differences to the perceived quality of the LLM's output (Kim et al., 2024), and recent work on instructional distraction shows that models can be sensitive to how task instructions are embedded in surrounding text (Hwang et al., 2025). To our knowledge, however, no prior work has systematically

studied which *interaction strategies* between human and AI lead to the best outcomes in the field of theoretical physics.

Our motivation is threefold. First, physicists are already consulting LLMs for their day-to-day calculations, meaning one should evaluate not only the first answer, but the interaction as a whole. Moreover, one should ask how reliably agent responses converge, how they respond to challenge, and where they fail. This is prerequisite to calibrating that usage in a regime not assessed by single-turn benchmarks (Chung et al., 2025; Gao et al., 2025; Zhang et al., 2025a). Second, theoretical research is moving towards workflows in which the physicist supervises a collection of AI agents, rather than interacting with one model at a time. In our automated benchmark this supervisory role is stylized as an external Judge, while moment-to-moment Critic feedback is delegated to an AI teaching assistant. In open-ended use, the physicist may instead occupy part of the Critic role directly by probing claims, supplying consistency checks, and deciding when the exchange has met the required standard. Understanding which scaffolding styles help which AI actors reach correct solutions is therefore a prerequisite for making either supervisory mode productive. Third, SCALAR gives us a controlled testbed in which to hypothesis-test widely repeated prompt-engineering claims — *e.g.*, the report that "assigning the model a persona" can swing performance by tens of percentage points (Gupta et al., 2024) — on current-generation models and on reasoning-heavy scientific tasks.

To study these questions we introduce independent axes of variation for each party. The Actor is varied through an *Actor persona*, defined as the combination of an *expertise level* (novice, expert, or unspecified default) and a *reasoning style* (meticulous, physical, skeptical, or left unspecified). This is the kind of Actor pre-prompting a physicist might use when asking a model to approach a problem in a particular way. The Critic is varied across a range of *Critic feedback strategies* (from lenient and pedagogical to strict and adversarial, plus an unspecified default), capturing how the assistant intervenes. During the dialogue, the Judge enters only as a reference-backed evaluator of correctness, silent with respect to the Actor–Critic exchange. Stored transcripts can then be re-scored by additional Judges, letting us separate interaction effects from Judge-specific scoring effects. This defines the role of the Judge as an authority that sets the standard of the exchange without actively participating in it. Each persona–strategy configuration is sampled several times on graduate-level QFT and string theory problems, allowing us to estimate trends across repeated dialogues rather than relying on single runs. In addition to endpoint scores and convergence rates, we use per-turn *score-update curves* as a compact diagnostic of when Critic feedback continues to move the Actor and when the dialogue's effect on the score has plateaued.

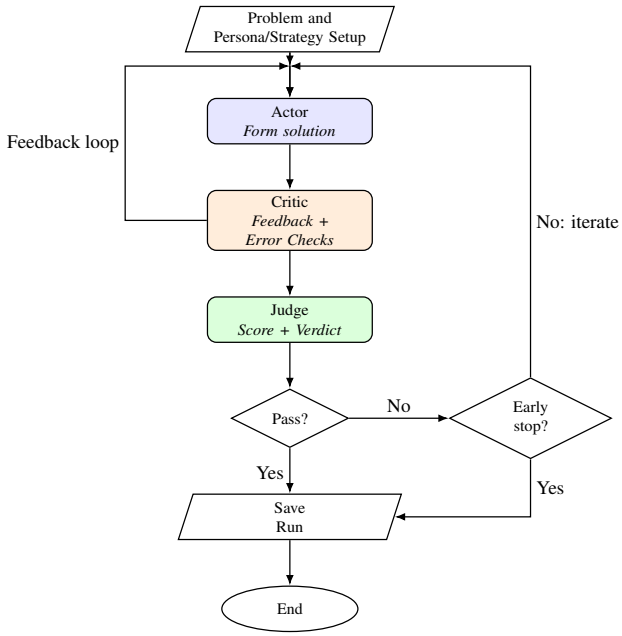

Figure 1. The SCALAR Actor–Critic–Judge pipeline. The Actor and the Critic engage in iterative dialogue, while an independent evaluator (Judge) scores the Actor's current solution against a ground truth. The Actor's output is shaped by the Actor persona; the Critic's feedback is shaped by the Critic feedback strategy.

While SCALAR is set up in a theoretical physics context, it can be straightforwardly extended to other domains. The lessons we are attempting to extract about LLM pre-prompting and interaction can eventually inform the optimization of multi-agent setups, where the agent persona and skill set play an important role. More generally, we view our analysis as a step towards more efficient AI-assisted open-ended research; the interaction patterns identified here provide a vocabulary for that future work.

2. Methods

2.1. Roles, Feedback Strategies and Pipeline

We begin the discussion of our methods by describing the SCALAR Actor personas and Critic feedback strategies, both of which are implemented through pre-prompting. The Actor persona is factored into two orthogonal dimensions. The first sets the expertise level: *expert* ("you are an expert in theoretical physics"), *novice* ("you are a student learning QFT"), or *default* (no expertise instruction). The second covers different reasoning styles, which shape the approach the Actor takes in the calculations: *meticulous* (emphasizing careful algebra and cross-checks), *physical* (prioritizing physical intuition and limiting cases), *skeptical* (questioning assumptions at each step), or *default* with no style instruction. The full set of combinations yields $3 \times 4 = 12$ Actor personas.

The Critic's feedback strategy controls the tone of the feedback provided to the Actor: *adversarial* (aggressively challenging claims), *strict* (precise error flagging), *pedagogical* (Socratic questioning), *lenient* (gentle suggestions accepting partial progress), and *default* with no stylistic emphasis. The full prompt texts are given in Appendix A. The LLM assigned to each role is an additional degree of freedom and our model choices are described in Section 2.3.

With the pre-prompting fixed, SCALAR proceeds as follows (Figure 1). Given a problem statement and persona–strategy configuration, the Actor produces an initial solution attempt. The Critic, who has access to a reference solution but is instructed not to disclose it, then reviews this attempt, flags errors, and delivers structured feedback. The Judge scores the Actor's work against the reference solution and issues a pass/fail verdict. If the Actor passes, or if an early-stopping criterion is met (iteration limit or score stagnation), the run is saved and terminated. Otherwise the Critic's feedback is passed back to the Actor for a further attempt and the loop repeats until the stopping criteria are met. For analysis, the recorded state after each Actor turn consists of the fixed experimental settings and the dialogue record available at that point. Thus the recorded dialogue state is the natural Markov state of the generated dialogue: conditional on that state and the fixed configuration, the next role call is generated without invoking any additional recorded dialogue history. In Section 3, we analyze the Judge score as a scalar projection of this evolving Markov state.

2.2. Evaluation and Metrics

At each iteration t the Judge scores the Actor's current solution in six dimensions totaling 100 points: correctness (50), mathematical rigor (10), logical flow (10), justification quality (10), completeness (10), and physical consistency (10). Let $s_t \in [0, 100]$ denote the total score at turn t. Here lowercase t indexes turns, while uppercase T_i denotes the number of scored Actor states in run i. A run with T scored Actor states produces a sequence $s_0, s_1, \ldots, s_{T-1}$. The Judge operates outside the Actor–Critic loop: its scores do not feed back into the dialogue, so the same transcripts can be re-scored by different Judge LLMs to separate dialogue-level effects from judge-specific ones. Below, a subscript i denotes a run, and angle brackets denote arithmetic averages over the indicated set of runs.

We report three evaluation metrics: two score-based quantities measured in points out of 100, and one rate that we report as a percentage of runs:

- **Mean per-turn score:** $\bar{s}_i = T_i^{-1} \sum_{t=0}^{T_i-1} s_{i,t} \in [0, 100]$. This is the average per-turn score across the whole dialogue. When we quote a group mean $\bar{s}$, we mean the arithmetic mean of these run-level quantities. When we instead quote a *final score*, we say so explicitly and mean s_{i,T_i-1} averaged over runs.
- **Gain:** $g_i = s_{i,T_i-1} - s_{i,0} \in [-100, +100]$. This is the endpoint improvement in score points out of 100. $g_i > 0$ means the dialogue made the solution better.
- **Convergence rate**: Let $r_i \in \{0, 1\}$ denote whether the run converged. A run is counted as converged when at least one iteration of the dialogue produces an Actor solution satisfying all three criteria: correctness ≥ 40 (*i.e.*, $\geq 80\%$ of the 50-point correctness rubric), total Actor score ≥ 80, and final-answer equivalence with the reference, under the scoring Judge. For runs whose original loop was driven by the same Judge that scores them, the passing iteration is also the terminal iteration; for re-scored transcripts the loop length is fixed by the original Judge and the rescoring Judge can mark a non-terminal iteration as passing. For any group of runs G, the convergence rate is $R_G = \langle r_i \rangle_{i \in G}$ and is reported as a percentage. See Appendix D for the formal rule.

For cross-problem Critic feedback strategy comparisons we also use *problem-normalized contrasts*. Let $m(i)$, $p(i)$, and $c(i)$ denote run i's Actor model setting (Haiku, DS8B, or DS70B), problem, and Critic feedback strategy. For Actor model setting m and Critic feedback strategy c, define

$$D_{\bar{s}}(m, c) = \left\langle \bar{s}_i - \langle \bar{s} \rangle_{m,p(i)} \right\rangle_{m(i)=m,\, c(i)=c},$$
$$D_R(m, c) = 100 \left\langle r_i - \langle r \rangle_{m,p(i)} \right\rangle_{m(i)=m,\, c(i)=c}.$$

Thus $D_{\bar{s}}$ is measured in score points and D_R in percentage points. These quantities are descriptive contrasts, not new raw scores: they ask whether a Critic feedback strategy sits above or below the local Actor–problem baseline. Here "local" means the baseline for the same Actor model setting and the same problem, after pooling over personas and Critic feedback strategies; the contrast therefore removes the much larger differences in baseline problem difficulty before comparing Critic feedback strategies.

2.3. Problems and Models

We test SCALAR on three graduate-level QFT and string theory problems drawn from standard textbooks. *Peskin 2.3* (Peskin & Schroeder, 1995) requires computing the Feynman propagator at spacelike separation, yielding the modified Bessel function result $\frac{m}{4\pi^2 r} K_1(mr)$. *Peskin 4.2* (Peskin & Schroeder, 1995) asks for the lowest-order lifetime of a heavy scalar particle decaying into two lighter scalars. *Polchinski 2.7* (Polchinski, 1998) requires deriving Operator Product Expansion (OPE) coefficients in the free boson CFT, a core calculation in conformal field theory. Each of these questions was specifically chosen to examine different facets of physics reasoning. For example, Peskin

2.3 and Polchinski 2.7 are conceptually straightforward, longer calculations, requiring care with algebraic manipulations, but less physical reasoning and intuition. Conversely, Peskin 4.2 is a technically shorter question designed to test conceptual knowledge of decay rates in QFT. For this work the problem set is intentionally chosen to be small, allowing us to deploy many strategies with modest compute. The tradeoff is that the results should be read as a controlled case study of interaction structure between agents rather than as a broad benchmark of theoretical-physics reasoning.

We study three Actor model settings and introduce the abbreviations used below. The first uses DeepSeek-R1 70B (DeepSeek-AI, 2025) (70B parameters; hereafter DS70B) as both the Actor and Critic, with QwQ-32B (Qwen Team, 2024) (32B parameters; hereafter QWQ) as the primary Judge; each of the $12 \times 5 = 60$ persona–strategy configurations is run with 15 repeats (5 per problem) at temperature T=0.7 for up to 4 iterations, yielding $n = 900$ total runs. The second uses DeepSeek-R1-0528-Qwen3-8B (8B parameters; recorded in the logs as `deepseek-r1`; hereafter DS8B) with the same DS70B Critic and QWQ Judge, again yielding $n = 900$ runs. The third uses Claude Haiku 4.5 (Anthropic, 2024) (parameter count not public; hereafter Haiku) as the Actor and Claude Sonnet 4.6 (parameter count not public; hereafter Sonnet) as both Critic and primary Judge, with 5 repeats per persona–strategy cell on Peskin 2.3 ($n = 300$ runs), and partial coverage on Peskin 4.2 ($n = 83$) and Polchinski 2.7 ($n = 51$).

For Haiku, the reduced-coverage runs served as exploratory screens. They showed that Haiku converges on only 3% of Polchinski 2.7 cases under the Sonnet Judge, so we did not extend this problem to the full sweep. In what follows, we therefore report detailed Haiku Critic feedback strategy statistics primarily on the two Peskin & Schroeder problems, while retaining the available Polchinski runs for exploratory per-problem QWQ comparisons.

To separate effects rooted in the dialogue itself from those specific to a particular Judge (see *e.g.*, (Tan et al., 2025) for a discussion on Judge reliability) we additionally re-score transcripts with independent Judges. For the Haiku runs Sonnet is the primary Judge; we use closed-model Opus on a small sample as a calibration check on Sonnet/QWQ agreement. The Haiku transcripts are re-scored by QWQ and by DS70B, and the two DeepSeek-family Actor model settings are re-scored by DS70B alongside the QWQ primary Judge. Throughout the main results we report QWQ as the common scalable Judge across all three Actor model settings and treat DS70B re-scoring as a sensitivity check.

Since the tasks consist of classical theoretical physics problems which may be inspected manually, we review representative outlier transcripts and ambiguous Judge decisions, checking whether apparent failures reflect genuine physics errors, grading artifacts, or limitations of the model solution. The QWQ Judge produces scores that are broadly consistent with the manual review, and closer to the closed-model audit set than DS70B re-scoring, suggesting that the QWQ model may reliably act as the primary grader for the purposes of this study.

The analyzed corpus is compute-dense: across Actor, Critic, primary Judge, and re-scoring calls, it contains roughly 9.4×10^7 estimated token-equivalents reconstructed from archived prompt/response text, including about 6.2×10^7 tokens of judging. Exact provider-side token counters were not preserved, so these estimates are a scale-of-compute summary rather than billing totals. This complements broader single-turn studies such as TPBench, which spans 57 theoretical-physics problems and 10 models with five attempts per model/problem (Chung et al., 2025); SCALAR instead spends inference budget on factorial persona–strategy sweeps, iterative refinement, and Judge sensitivity over repeated transcripts. This comparison should not be read as a claim that unconstrained multi-agent dialogue is generically superior to single-turn evaluation: the SCALAR Critic is reference-conditioned, seeing the solution while being instructed not to disclose it, so the loop is closer to supervised tutoring than to open-ended autonomous discovery.

3. Results

We summarize performance using the evaluation metrics defined in Section 2.2: the mean per-turn score $\bar{s}$, the gain g, and the convergence rate R. When comparing strategies across problems, we additionally use the descriptive problem-normalized contrasts $D_{\bar{s}}$ and D_R. We refer to one complete Actor–Critic–Judge dialogue trajectory as a *run*; each run is treated as one independent observation. For statistical comparisons, we use: Wilcoxon signed-rank to investigate the sign and significance of paired turn-0/turn-$(T-1)$ gain differences; Kruskal–Wallis to detect differences across the five Critic feedback strategies; and Mann–Whitney for further pairwise Critic feedback strategy comparisons. These procedures use the raw run-level evaluation quantities, not the problem-normalized contrasts. Definitions and procedure details are given in Appendix D.

Structured feedback improves first attempts. For all three Actor model settings, the final solution is substantially better than the Actor's first attempt under the common QWQ scoring (Figure 2). On the two Peskin problems, Haiku with a Sonnet Critic reaches high QWQ convergence (87.7% on Peskin 2.3 and 96.4% on Peskin 4.2), while the reduced Polchinski screen remains much harder.[1] The available re-

[1] In the exploratory Polchinski 2.7 Haiku screen, QWQ convergence is 27.5% ($n = 51$).

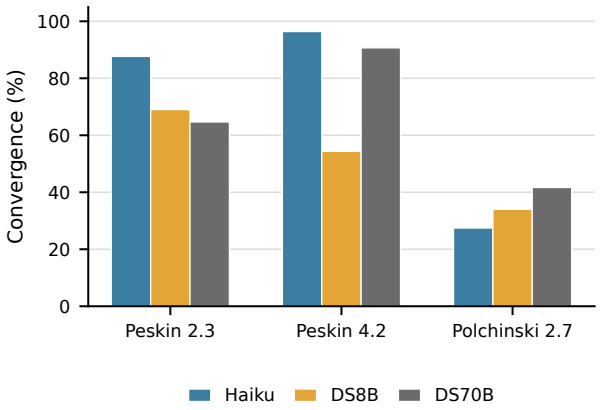

Figure 2. Per-problem convergence for the three Actor model settings. Haiku uses the Sonnet Critic; DS8B and DS70B both use the DS70B Critic. Haiku has reduced coverage on Peskin 4.2 and Polchinski 2.7 ($n = 83$ and $n = 51$, respectively), while the other shown cells use $n = 300$ runs.

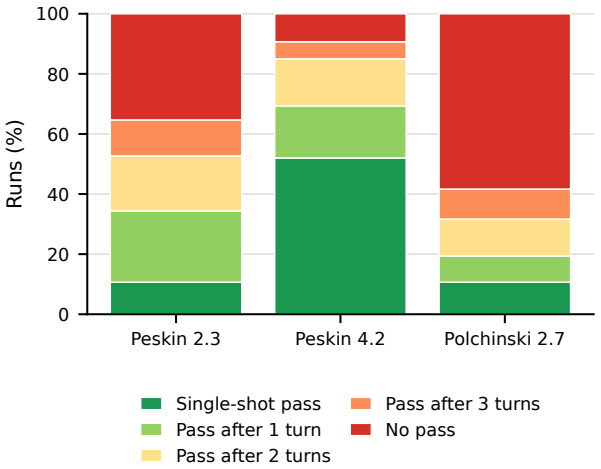

Figure 3. Run-fate breakdown by problem for the DS70B Actor under the QWQ Judge. Bars show single-shot passes, multi-turn passes after one/two/three Critic turns, and runs that terminate without passing. The large single-shot component on Peskin 4.2 reflects cases where the Actor passes before Critic feedback is needed.

scoring checks preserve the sign of the multi-turn gain, so this conclusion is not an artifact of the common QWQ scale.

For DS70B on all three problems, QWQ convergence reaches 65.7% (n=900), with mean final score 80.7 and mean gain $g = +13.4$. DS8B reaches lower QWQ convergence, 52.4% (n=900), with mean final score 76.6 and mean gain $g = +13.3$. This within-family comparison is useful because the Critic is held fixed: DS8B resembles DS70B more than Haiku in its Critic feedback strategy sensitivity, even though its absolute performance is lower under QWQ. Under DS70B re-scoring the within-family ordering becomes judge-sensitive, so the common figures use QWQ rather than treating the DS8B/DS70B scale ordering as a judge-independent claim.

The DS70B turn-0 score under QWQ averages 67.3 and climbs to 80.6 by the end of the Critic loop, closing roughly 40% of the gap to saturation despite the stronger single-shot baseline. This improvement is not a product of prompt optimization alone: it requires the iterative structure, consistent with structured multi-agent refinement reducing errors (Till et al., 2025) and contrasting with the failure of naive sequential strategies (Gao et al., 2025) and with prior findings that single-turn prompt optimization yields no significant gain (Chung et al., 2025).

The three problems span a clear difficulty spectrum for DS70B, which manifests as a shift in the mixture of run-fate categories (Figure 3). QWQ convergence is 90.7% on Peskin 4.2, 64.7% on Peskin 2.3, and 41.7% on Polchinski 2.7, with $n = 300$ runs per problem. DS8B does not follow the same problem ordering: QWQ convergence is 69.0% on Peskin 2.3, 54.3% on Peskin 4.2, and 34.0% on Polchinski 2.7, with $n = 300$ runs per problem (Figure 2). Polchinski 2.7 remains the hardest problem across all three Actor model settings, but DS8B finds Peskin 2.3 easier than Peskin 4.2 — the opposite of what DS70B and Haiku do. The same reversal appears in the score-update curves discussed next: for DS8B, Peskin 4.2 stops improving around intermediate scores, whereas Peskin 2.3 continues to improve until higher scores.

Score-update curves describe dialogue dynamics. The per-problem convergence rates say where the Actor–Critic loop ends, but not how later Critic turns move the solution. Because SCALAR records the fixed configuration and dialogue state after each turn (Section 2.1), that recorded state is the natural Markov state of the generated dialogue. The Markovian interpretation first becomes interesting when we project that state to a single observable: the Judge score. For the DeepSeek-family comparison in Figure 4, we estimate the projected *score-update field*

$$v(s) = \mathbb{E}[s_{t+1} - s_t \mid s_t \simeq s]\,.$$

The average is over all observed next-turn transitions in the same score bin for a fixed Actor model and problem, pooling over personas and Critic strategies. A zero of this curve is a *projected fixed point*: the next Critic turn no longer improves the Actor on average in the observed ensemble near that score. Throughout the rest of the paper, "fixed point" refers to this projected score-level object, not to a complete fixed point of the full recorded dialogue state. This is the quantity that separates "already easy," "still improvable," and "apparently stuck" regimes that are compressed together by endpoint gain. In Appendix E, the gain distribu-

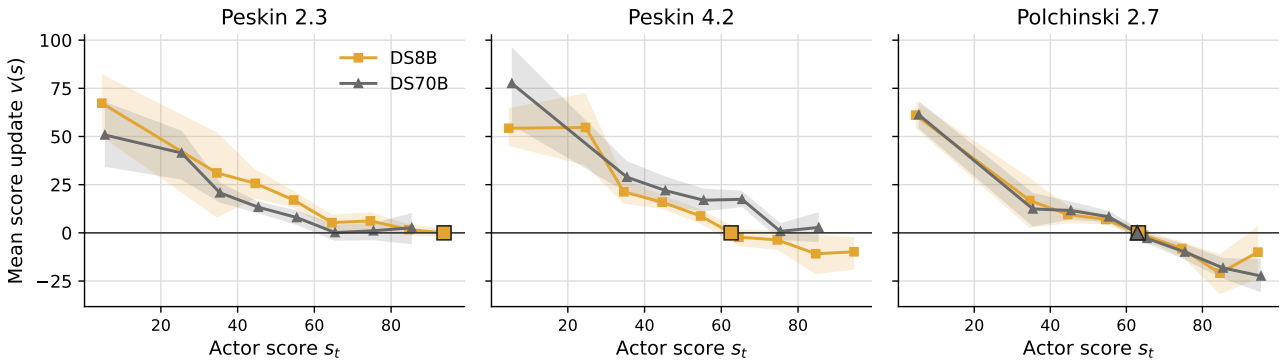

Figure 4. Empirical score-update curves for the two DeepSeek-family Actor scales under the common QWQ scoring. Points estimate the projected score-update field $v(s) = \mathbb{E}[\Delta s_t \mid s_t \simeq s]$ by averaging next-turn updates $\Delta s_t = s_{t+1} - s_t$ in bins of the current Actor score s_t for the indicated Actor model and problem, pooling over personas and Critic feedback strategies; shaded bands are bootstrap 95% confidence intervals. Only runs with at least one observed update contribute, so immediate single-shot passes are not part of these curves. Markers on the zero line indicate estimated fixed points s^*, meaning zeros of the projected score-update field.

tions show the same point from a different angle: endpoint gain mixes first-shot successes, rescued runs, and stuck trajectories rather than isolating a clean strategy effect.

Recent work has imported dynamical-systems language into LLM behavior: Carson (2025) model sentence-level reasoning trajectories of open-source LLMs as a switching linear dynamical system on a latent manifold, Sarfati et al. (2025) characterize token-level activation paths as approximately linear-drift stochastic processes, and Wang et al. (2025) study attractor cycles under repeated paraphrasing. SCALAR puts related language to work in a different setting: the underlying state is a multi-agent dialogue, the driving field is supplied by a reference-conditioned Critic, and the observable is an externally judged physics score. The result is a score projection of a Markov transcript process. Appendix E states the corresponding projection and transition-kernel definitions, and reports residual checks that support the score projection as a useful coarse-grained diagnostic while leaving predictive kernel modeling to larger studies.

Under this view, the within-family comparison reveals different regimes. On Peskin 2.3 both DeepSeek-family Actors continue to improve until they reach high scores. On Peskin 4.2, DS8B develops a mid-score fixed point near $s_t \simeq 63$, whereas DS70B remains in a positive-drift regime through most of the observed range and only crosses near high score; we therefore do not mark a clean mid-score fixed point for DS70B on this problem. On Polchinski 2.7, however, both DS8B and DS70B have a similar fixed-point region near $s_t \simeq 63$. In this sense, increasing parameter count within the same model family changes the easier-problem dynamics but does not remove the hard-problem bottleneck.

Critic feedback strategy is model dependent. Feedback style should matter most when the Critic is actively moving the Actor within a dialogue, rather than when a run passes immediately or remains stuck. This is what we observe for Haiku, but not as a stable single-model effect for either DeepSeek Actor scale.

Under the common QWQ scoring, Haiku is more sensitive to Critic feedback strategy than either DeepSeek Actor scale (Figure 5). The normalization is needed because raw mean score and raw convergence are dominated by problem difficulty; the contrasts instead isolate within-problem Critic feedback strategy shifts. Under this view, Haiku places pedagogical, lenient, and default feedback above its local baseline in $D_{\bar{s}}$, with strict and adversarial below it. This visual pattern matches the raw $\bar{s}$ Critic feedback strategy test for Haiku (Kruskal–Wallis p=0.012), while the corresponding convergence-rate test is not significant (p=0.69). Thus the Haiku Critic feedback strategy signal is clearest in mean per-turn score and should not be read as a strict ranking of the five strategies.

For DS70B alone the five Critic feedback strategies are statistically indistinguishable. Across the 900-run sweep, the raw QWQ $\bar{s}$ means by Critic feedback strategy span only a few score points, and the omnibus Critic-feedback-strategy tests are null (Kruskal–Wallis p=0.61 for $\bar{s}$ and p=0.17 for R under QWQ). DS70B re-scoring gives the same null conclusion. DS8B is similar: under QWQ, the omnibus tests find no reliable Critic feedback strategy effect on $\bar{s}$ (p=0.10) or convergence rate R (p=0.22), and DS70B re-scoring agrees. Descriptively, however, both DeepSeek-family Actors put lenient feedback first under QWQ. Naively pooling the two DeepSeek Actor scales yields a weak lenient advantage for raw $\bar{s}$ and convergence rate. We therefore report this as a QWQ-conditioned tendency, not as the same kind of robust Critic feedback strategy effect seen for Haiku. Across

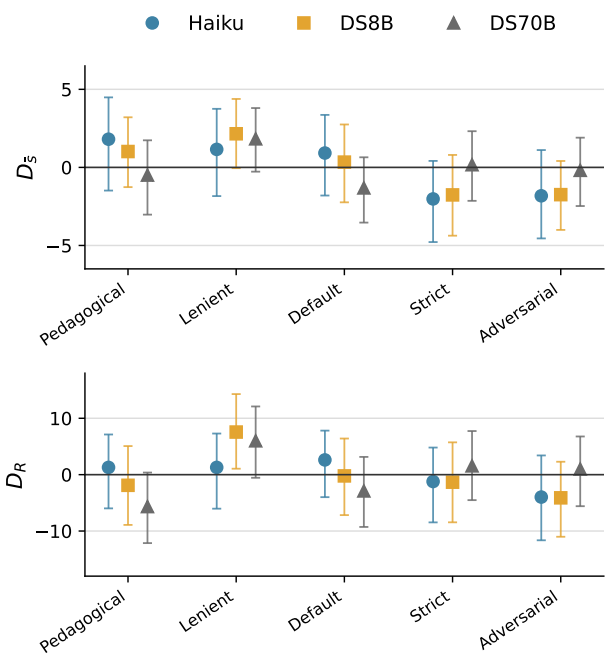

Figure 5. Problem-normalized Critic feedback strategy contrasts under QWQ scoring. Haiku values use the two Peskin problems (n=383); DS8B and DS70B values use all three problems (n=900 each). For each Actor model setting and problem, we subtract the local baseline before averaging by Critic feedback strategy, so the figure asks which Critic feedback strategies sit above or below the corresponding model–problem baseline rather than which problems are easiest. Vertical bars show descriptive run-level bootstrap 95% confidence intervals after problem-centering. $D_{\bar{s}}$ is measured in score points on the 0–100 Judge scale; D_R is measured in percentage points of convergence.

all three Actor model settings, we find no stable evidence that adversarial or strict feedback is the best Critic feedback strategy.

Actor persona prompting has negligible effect. Actor persona prompts are even weaker as a design variable. For DS70B under the common QWQ scoring, the 12 Actor-persona means span only $\bar{s} \in [69.4, 74.1]$ on the full 900-run sweep — a 5-point range across 12 configurations, no larger than what sampling variation alone would produce if all configurations had identical effects (Kruskal–Wallis p=0.99). Varying the Actor persona along expertise and reasoning-style axes has no measurable effect on DS70B outcomes. DS8B gives the same broad message: the few low-p axis tests are not stable across evaluation metrics or re-scoring checks, while the expertise axis itself is null.

Appendix A shows the full 60-cell persona×strategy heatmaps for the balanced DeepSeek-family sweeps, supporting the same conclusion: no single Actor persona is consistently bright across problems, models, and Critic strategies.

For Haiku under QWQ scoring, the expertise axis also does not robustly beat the unspecified default once we restrict to the two Peskin & Schroeder problems ($\bar{s}$: default 75.5 vs expert 74.1). The available re-scoring checks do not turn this into a stable dialogue-level expertise effect. Taken together with the Critic feedback strategy analysis above, the lesson is that among the prompt axes we tested only Critic feedback strategy reliably moves Haiku mean-score results, while neither DeepSeek Actor scale shows a robust single-model persona or Critic feedback strategy effect. Actor persona prompting is therefore not a reliable design variable in either pairing.

4. Discussion and Outlook

4.1. Practical Takeaways

Two observations from our experiments may be useful when deploying SCALAR for graduate-level physics reasoning.

The Actor–Critic pairing is the key design choice. Critic feedback strategy matters, but not universally: it must be tuned for the Actor–Critic system rather than chosen as a general rule. The same multi-turn structure improves all three Actor model settings, but the mechanism differs: Haiku improves smoothly within the dialogue, whereas the DeepSeek-family averages combine first-shot successes, rescued runs, and failures that remain stuck. For a physicist using a frontier assistant, this gives three practical lessons:

1. Use dialogue rather than only a one-shot request.[2] Across all Actor model settings, multi-turn refinement improves over the initial solution, especially when later turns target concrete physics checks such as dimensions, limits, symmetries, known special cases, and missing factors.

2. Do not expect Actor persona prompts to carry the result. Pre-prompting current reasoning Actors with stronger personas, such as "be an expert theorist," is not a reliable design variable in our experiments.

3. Treat Critic feedback strategy as the prompt lever most worth testing, but do not assume that harsher criticism is better. Strict and adversarial feedback are never stably best in our experiments. Among the axes studied here, this is where we see the clearest model-dependent movement in mean score. A better starting point is constructive feedback that preserves correct partial work, targets the missing physics check, and only then tightens the standard; in the DeepSeek-family

[2]This recommendation is hardly revolutionary, but it is often the missing step in practice: a single flawed response is still sometimes treated by physicists as a verdict on the model, rather than as the first turn of an interaction.

settings, the lenient signal is suggestive but remains QWQ-conditioned rather than a robust prescription.

Furthermore, recent results that support the importance of asymmetric Actor-Critic pairing include (Jiang et al., 2026). This also suggests that prompt-engineering folklore deserves re-testing on current reasoning-tuned models, especially in dialogue rather than only in single-turn settings.

Dialogue dynamics are regime dependent. Recent controlled studies of reasoning models report complexity-dependent regimes, including abrupt performance collapse beyond critical task difficulty and phase-transition-like behavior in logical reasoning benchmarks (Shojaee et al., 2025; Hazra et al., 2025; Zhang et al., 2026). Our three physics problems are not a controlled complexity ladder, so we do not claim a phase transition.

Instead, the score-update analysis in Section 3 gives a conservative way to describe dialogue regimes: easy instances often pass before Critic feedback, intermediate instances can be moved by structured feedback, and hard instances can remain stuck despite additional dialogue. The DS8B/DS70B comparison is a first controlled scale probe within one model family: increasing parameters changes the easy/intermediate regimes, but does not by itself remove the Polchinski bottleneck. A denser DS-R1 scale ladder could test whether such fixed-point regions disappear at problem-dependent critical scales; with only two scales here, this remains motivated future work rather than a scaling claim.

In Appendix C, Figure 8 also identifies justification quality as the most persistent rubric gap. This suggests a natural next step for SCALAR: treat Actor–Critic exchanges as controlled dynamical systems, vary model scale, Critic strength, turn budget, and problem difficulty, and test whether empirical transition kernels predict held-out convergence. A richer state could also include Critic scores, turning the one-dimensional Actor-score projection into a joint description of feedback quality and Actor uptake.

This connects to the "agentic gap" critique of static text-only reasoning evaluations (Khan et al., 2025): scaffolding can rescue some failures, but our hard-problem results show that scaffolding itself has boundaries.

4.2. Toward AI-Assisted Scientific Discovery

AI-assisted discovery in fundamental physics is moving from speculation to an operational research question (Lu et al., 2025). Recent Gemini case studies give an open-ended perspective: frontier models can be useful in scientific workflows when embedded in human-supervised loops involving decomposition, critique, code generation, and external validation (Woodruff et al., 2026). SCALAR takes the complementary controlled direction: by returning to known-reference problems, it makes one part of that workflow measurable, namely how Critic feedback changes the trajectory of an Actor's reasoning.

At the model level, one might expect that fine-tuning models on domain-specific high-energy theory corpora, as in the FeynTune study (Richmond et al., 2026), would substantially improve research-style reasoning in high-energy theory. However, the results of that study indicate that targeted fine-tuning at the scales currently accessible to small academic collaborations does not close the gap with frontier general-purpose models. This motivates the SCALAR approach: rather than trying to compete with frontier capability at the model level, scaffold a frontier model with a structured Critic and obtain additional gains at the *interaction* level. We test this idea on graduate-level QFT and string-theory calculations in a controlled benchmark.

However, in the present setup, the Critic is reference-conditioned: it has access to reference solutions and therefore guides the Actor with privileged knowledge of the answer. The Judge, by contrast, represents the external-validation part of the physicist's supervisory role: it evaluates the Actor–Critic exchange from outside the dialogue and determines whether the final reasoning meets the required standard. This is appropriate for benchmarking, but it means that the grading system still presupposes a known solution.

The longer-term vision is to remove this dependence on reference solutions. [3] For open problems, the researcher would instead move into the Critic role, supplying consistency checks and domain intuition. The Judge would then be replaced by an external arbiter — experimental measurements, simulation outputs, theorem-prover checks, or mathematical consistency conditions — so that the dialogue is validated against available ground truth rather than against a textbook solution. Appendix C explains which parts of the rubric would survive, and which would need replacement, in such a no-reference setting. In this setting, the three-agent analogy with the scientific method would become literal rather than stylized. Understanding how to configure Actor and Critic so that their dialogue reliably converges on problems whose answers are known is therefore a prerequisite for the more ambitious goal of collaborating on problems whose answers are not.

[3] This connects to a broader program in AI-assisted science: models can already learn useful structure from scientific data, but extracting reliable human-usable insight from that structure often remains labor intensive (Schmidt & Lipson, 2009; Udrescu & Tegmark, 2020; Cranmer et al., 2020; Kitouni et al., 2024; Richardson et al., 2025). Agentic workflows could eventually help automate parts of this interpretive loop.

References

Agrawal, P., Craig, N., Madden, A., and Valenzuela Lombera, I. The FERMIACC: Agents for particle theory. *arXiv preprint arXiv:2603.22538*, 2026.

Anthropic. The Claude Model Card and System Prompt, 2024. https://www.anthropic.com.

Carson, T. A statistical physics of language model reasoning. *arXiv preprint*, 2025.

Chung, D. et al. Theoretical physics benchmark (TPBench): A dataset and study of AI reasoning capabilities in theoretical physics. *arXiv preprint arXiv:2502.15815*, 2025.

Cranmer, M., Sanchez-Gonzalez, A., Battaglia, P., Xu, R., Cranmer, K., Spergel, D., and Ho, S. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17429–17442, 2020.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2024.

Estornell, A., Ton, J.-F., Yao, Y., and Liu, Y. ACC-Collab: An actor-critic approach to multi-agent LLM collaboration. *arXiv preprint arXiv:2411.00053*, 2024.

Gao, Z. et al. Test-time scaling techniques in theoretical physics: A comparison of methods on the TPBench dataset. *arXiv preprint*, 2025.

Guevara, A., Lupsasca, A., Skinner, D., Strominger, A., and Weil, K. Single-minus gluon tree amplitudes are nonzero. *arXiv preprint arXiv:2602.12176*, 2026.

Gupta, J. et al. Persona is a double-edged sword: Enhancing the zero-shot reasoning by ensembling the role-playing and neutral prompts. *arXiv preprint arXiv:2408.08631*, 2024.

Hazra, R., Venturato, G., Zuidberg Dos Martires, P., and De Raedt, L. Have large language models learned to reason? a characterization via 3-SAT phase transition. *arXiv preprint arXiv:2504.03930*, 2025.

Hwang, Y., Kim, Y., Koo, J., Kang, T., Bae, H., and Jung, K. LLMs can be easily confused by instructional distractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19483–19496. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.957. URL https://aclanthology.org/2025.acl-long.957/.

Jaiswal, R. et al. Improving physics reasoning in large language models using mixture of refinement agents. *arXiv preprint arXiv:2412.00821*, 2024.

Jiang, S., Zhang, Z., Zhang, Y., Yang, S., Xia, W., and Soatto, S. Asymmetric actor-critic for multi-turn llm agents, 2026. URL https://arxiv.org/abs/2604.00304.

Khan, S., Madhavan, S., and Natarajan, K. A comment on "the illusion of thinking": Reframing the reasoning cliff as an agentic gap. *arXiv preprint arXiv:2506.18957*, 2025.

Kim, J., Yang, N., and Jung, K. Persona is a double-edged sword: Mitigating the negative impact of role-playing prompts in zero-shot reasoning tasks, 2024. URL https://arxiv.org/abs/2408.08631.

Kitouni, O., Nolte, N., Pérez-Díaz, V. S., Trifinopoulos, S., and Williams, M. From neurons to neutrons: A case study in interpretability. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 24726–24748. PMLR, 2024. URL https://proceedings.mlr.press/v235/kitouni24a.html.

Kruskal, W. H. and Wallis, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, 1952. doi: 10.1080/01621459.1952.10483441.

Laban, P. et al. LLMs get lost in multi-turn conversation. *arXiv preprint*, 2025.

Liang, Z., Yu, D., Yu, W., Yao, W., Zhang, Z., Zhang, X., and Yu, D. Mathchat: Benchmarking mathematical reasoning and instruction following in multi-turn interactions, 2024. URL https://arxiv.org/abs/2405.19444.

Lu, S., Jin, Z., Zhang, T. J., Kos, P., Cirac, J. I., and Schölkopf, B. Can theoretical physics research benefit from language agents? *arXiv preprint arXiv:2506.06214*, 2025.

Mann, H. B. and Whitney, D. R. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1): 50–60, 1947. doi: 10.1214/aoms/1177730491.

Novikov, A., Vu, N., Eisenberger, M., Dupont, E., Huang, P.-S., Wagner, A. Z., Shirobokov, S., Kozlovskii, B., Ruiz, F. J. R., Mehrabian, A., Kumar, M. P., See, A., Chaudhuri, S., Holland, G., Davies, A., Nowozin, S., Kohli, P., and Balog, M. AlphaEvolve: A coding agent for scientific and

algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Peskin, M. E. and Schroeder, D. V. *An Introduction to Quantum Field Theory*. Addison-Wesley, 1995.

Plehn, T., Schiller, D., and Schmal, N. MadAgents. *arXiv preprint arXiv:2601.21015*, 2026.

Polchinski, J. *String Theory*, volume 1. Cambridge University Press, 1998.

Qwen Team. QwQ: Reflect Deeply on the Boundaries of the Unknown, 2024. https://qwenlm.github.io/blog/qwq-32b-preview/.

Richardson, K. A., Trifinopoulos, S., and Williams, M. The dna of nuclear models: How ai predicts nuclear masses. *arXiv preprint arXiv:2508.08370*, 2025. URL https://arxiv.org/abs/2508.08370.

Richmond, P., Papageorgakis, C., Niarchos, V., Chowdhury, B., and Agarwal, P. FeynTune: large language models for high-energy theory. *Mach. Learn. Sci. Tech.*, 7(2): 025012, 2026. doi: 10.1088/2632-2153/ae47bb.

Romera-Paredes, B. et al. Mathematical discoveries from program search with large language models. *Nature*, 625: 468–475, 2024.

Sarfati, R. et al. Lines of thought in large language models. In *ICLR*, 2025.

Schmidt, M. and Lipson, H. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

Schwartz, M. D. Resummation of the C-Parameter Sudakov Shoulder Using Effective Field Theory. *arXiv preprint arXiv:2601.02484*, 1 2026.

Shih, D. Learning to Unscramble Feynman Loop Integrals with SAILIR. *arXiv preprint arXiv:2604.05034*, 4 2026a.

Shih, D. Learning to Unscramble: Simplifying Symbolic Expressions via Self-Supervised Oracle Trajectories. *arXiv preprint arXiv:2603.11164*, 3 2026b.

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., and Farajtabar, M. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941*, 2025.

Tan, S., Zhuang, S., Montgomery, K., Tang, W. Y., Cuadron, A., Wang, C., Popa, R. A., and Stoica, I. Judgebench: A benchmark for evaluating llm-based judges, 2025. URL https://arxiv.org/abs/2410.12784.

Till, R. et al. Multi-model consistency improves hallucination detection and mitigation in large language models. *arXiv preprint*, 2025.

Udrescu, S.-M. and Tegmark, M. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Vygotsky, L. S. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978. Edited by Michael Cole, Vera John-Steiner, Sylvia Scribner, and Ellen Souberman.

Wang, Y. et al. Unveiling attractor cycles in large language models: A dynamical systems view of successive paraphrasing. *arXiv preprint*, 2025.

Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945. doi: 10.2307/3001968.

Wood, D., Bruner, J. S., and Ross, G. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976. doi: 10.1111/j.1469-7610.1976.tb00381.x.

Woodruff, D. P., Cohen-Addad, V., Jain, L., Mao, J., Zuo, S., et al. Accelerating scientific research with gemini: Case studies and common techniques. *arXiv preprint arXiv:2602.03837*, 2026. URL https://arxiv.org/abs/2602.03837.

Xu, Y., Kimlee, H., Xiao, Y., and Luo, D. Advancing AI-scientist understanding: Multi-agent LLMs with interpretable physics reasoning. *arXiv preprint arXiv:2504.01911*, 2025. ICML 2025 Workshop on MAS.

Zhang, X., Dong, Y., et al. PhysReason: A comprehensive benchmark towards physics-based reasoning. *arXiv preprint arXiv:2502.12054*, 2025a.

Zhang, X., Zhang, Y., Chen, Z., Yu, J., Yang, W., and Song, Z. Logical phase transitions: Understanding collapse in LLM logical reasoning. *arXiv preprint arXiv:2601.02902*, 2026.

Zhang, Y. et al. TurnBench-MS: A benchmark for evaluating multi-turn, multi-step reasoning in large language models. *arXiv preprint*, 2025b.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances*

in Neural Information Processing Systems, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A. Prompt Texts

In SCALAR, Actor persona pre-prompting is factored into expertise × reasoning style. Representative summaries of each component follow; the full prompts used in experiments are provided in the code repository.

Expertise level. *Expert*: "You are an expert in theoretical physics with deep knowledge of QFT and mathematical methods." *Novice*: "You are a graduate student learning QFT, working through problems to build understanding." *Default (d)*: no expertise instruction.

Reasoning style. *Meticulous*: "Check every algebraic step carefully. Verify signs, prefactors, and limits before proceeding." *Physical*: "Prioritize physical intuition. Use dimensional analysis, limiting cases, and symmetry to guide your calculation." *Skeptical*: "Question every assumption. If a step seems unjustified, flag it and consider alternatives." *Default (d)*: no style instruction.

Feedback strategies. *Adversarial*: "Aggressively challenge every claim. Demand explicit justification for each step." *Strict*: "Flag every error precisely with the correct form." *Pedagogical*: "Guide through Socratic questioning. Ask leading questions rather than stating errors directly." *Lenient*: "Focus on what the solver got right. Offer gentle suggestions. Accept partial progress." *Default (d)*: no additional style instructions.

The heatmaps in Figures 6 and 7 show the balanced persona×strategy grids for the two DeepSeek-family sweeps. The block structure separates default, expert, and novice Actor personas. The visual pattern is irregular rather than block-diagonal: individual cells can be bright, but there is no stable persona formula that dominates across problems.

B. List of Physics Problems

As was outlined in Section 2.3, in this work we consider three problems chosen to probe various qualities of each model-prompt pair. Each problem is reproduced below.

Peskin 2.3 (Peskin & Schroeder, 1995) Evaluate the function

$$\langle 0|\phi(x)\phi(y)|0\rangle = D(x-y) = \int \frac{d^3p}{(2\pi)^3}\frac{1}{E_{\vec{p}}}e^{-ip\cdot(x-y)}$$

for $(x-y)$ spacelike so that $(x-y)^2 = -r^2$, explicitly in terms of Bessel functions.

Polchinski 2.7 (Polchinski, 1998) Consider the free CFT of free scalars X^μ.

1. By computing the relevant OPEs, confirm the following weights

$$\begin{aligned} &X^\mu \ \ (0,0)\\ &\partial X^\mu \ \ (1,0)\\ &\bar\partial X^\mu \ \ (0,1)\\ &\partial^2 X^\mu \ \ (2,0)\\ &:e^{ik\cdot X}: \ \ \left(\frac{\alpha' k^2}{4}, \frac{\alpha' k^2}{4}\right)\end{aligned}$$

   and determine which operators are tensors.

2. Do this for the same operators in the linear dilaton theory.

Peskin 4.2 (Peskin & Schroeder, 1995) Consider the following Lagrangian, involving two real scalar fields Φ and ϕ:

$$\mathcal{L} = \tfrac{1}{2}(\partial_\mu\Phi)^2 - \tfrac{1}{2}M^2\Phi^2 + \tfrac{1}{2}(\partial_\mu\phi)^2 - \tfrac{1}{2}m^2\phi^2 - \mu\Phi\phi\phi.$$

The last term is an interaction that allows a Φ particle to decay into two ϕ's, provided that $M > 2m$. Assuming that this condition is met, calculate the lifetime of the Φ to lowest order in μ.

C. Scoring Criteria

We evaluate each candidate solution using an *LLM Judge* that compares the submitted solution against the problem statement and, when available, a reference solution. The Judge produces a structured verdict consisting of binary pass/fail, fine-grained sub-scores, error flags, and qualitative summaries.

Inputs. For each evaluation instance, the Judge receives:

1. the problem statement P,
2. the Actor-generated solution A,
3. Critic feedback C embedded in the evaluated text, and
4. a reference solution R (when available).

Final-result equivalence check. Before assigning scores, the Judge performs an explicit *final-result comparison* between the Actor's claimed final answer and the reference answer. Let

$$\mathrm{Eq}(A,R) \in \{0,1\}$$

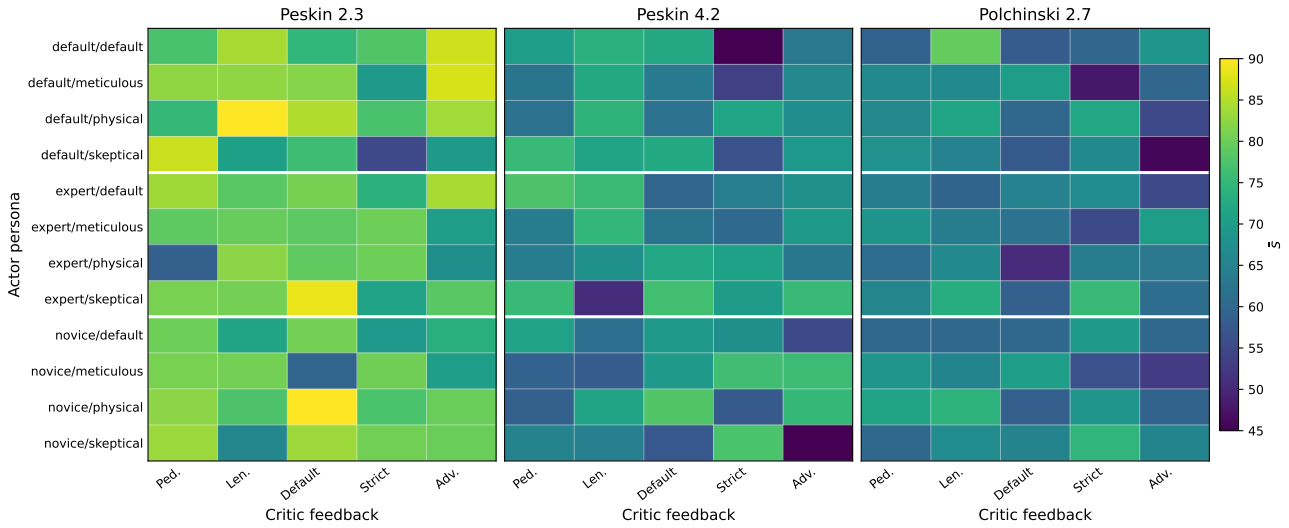

Figure 6. DS8B $\bar{s}$ heatmaps under QWQ scoring, separated by problem. Rows are Actor personas, columns are Critic feedback strategies, and cell color gives the mean per-turn score.

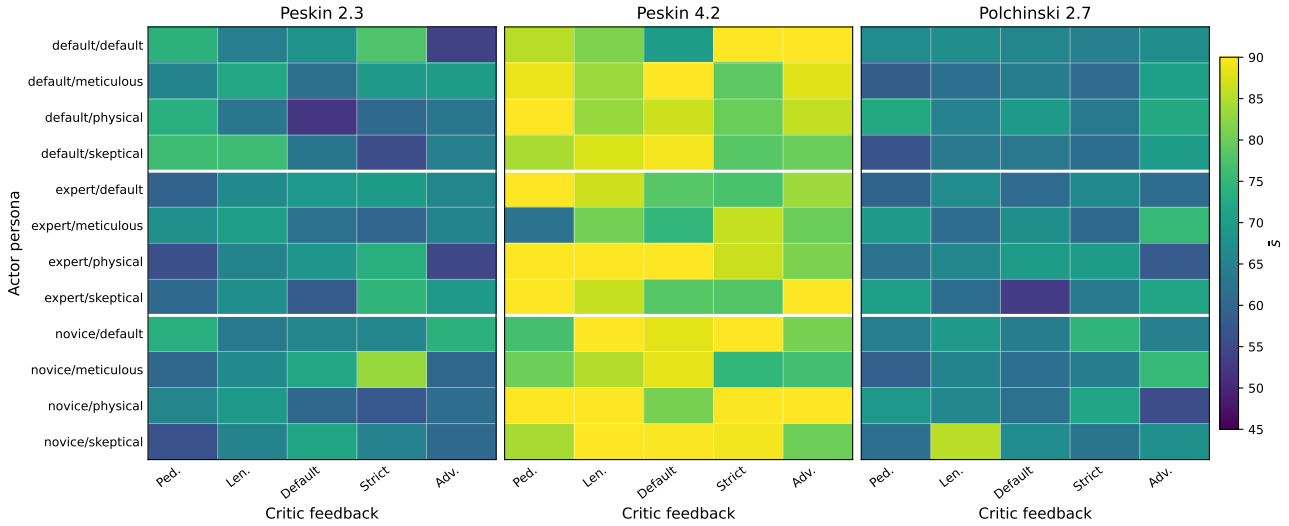

Figure 7. DS70B $\bar{s}$ heatmaps under QWQ scoring, separated by problem. Rows are Actor personas, columns are Critic feedback strategies, and cell color gives the mean per-turn score.

denote whether the Actor's final result is mathematically equivalent to the reference. Equivalence is defined up to:

- algebraic rearrangement,
- equivalent notation,
- trivial reordering of factors.

Non-equivalence is declared if the two results differ in sign, numerical prefactor, functional form, missing or extra terms, or complex phase factors. If the Actor provides no final formula or only an incomplete formula, then

$$\mathrm{Eq}(A, R) = 0.$$

Actor score. The Actor is evaluated along six dimensions:

$$S_{\text{actor}} = S_c + S_r + S_l + S_j + S_m + S_p, \qquad S_{\text{actor}} \in [0, 100],$$

where:

$S_c \in [0, 50]$	correctness,
$S_r \in [0, 10]$	mathematical rigor,
$S_l \in [0, 10]$	logical flow,
$S_j \in [0, 10]$	quality of justification,
$S_m \in [0, 10]$	completeness,
$S_p \in [0, 10]$	physical consistency.

Correctness (S_c). Correctness is based on the error density

$$\rho = \frac{N_{\text{errors}}}{N_{\text{steps}}}.$$

The rubric is:

- $S_c = 50$: $\rho = 0$, all equations are mathematically sound, and the final result is within 1% of the expected result;
- $S_c \in [42, 49]$: $\rho < 0.1$, only minor computational errors;
- $S_c \in [27, 41]$: $0.1 \le \rho < 0.3$, some correct intermediate reasoning but substantial gaps;
- $S_c \in [0, 26]$: $\rho \ge 0.3$, indicating fundamental conceptual or methodological errors.

Mathematical rigor (S_r). Rigor is measured by the justification ratio

$$J = \frac{N_{\text{justified}}}{N_{\text{statements}}}.$$

The rubric is:

- $S_r = 10$: $J = 1$, all statements justified;
- $S_r \in [7, 9]$: $J \ge 0.8$, mostly rigorous;
- $S_r \in [4, 6]$: $0.5 \le J < 0.8$, partial justification;
- $S_r \in [0, 3]$: $J < 0.5$, predominantly unjustified reasoning.

Logical flow (S_l). Logical coherence is assessed using the dependency structure of the derivation:

- $S_l = 10$: the derivation forms a clear directed acyclic graph (DAG), meaning a non-circular dependency structure in which each step follows from prior steps;
- $S_l \in [7, 9]$: mostly sequential with minor organizational gaps;
- $S_l \in [4, 6]$: unclear dependencies or partially disconnected reasoning;
- $S_l \in [0, 3]$: non-sequential reasoning, circularity, or disconnected argument structure.

Quality of justification (S_j). Explanation depth is measured by the average reasoning-chain length

$$R = \mathrm{avg}(\text{explanation steps per result}).$$

The rubric is:

- $S_j = 10$: $R \ge 3$;
- $S_j \in [7, 9]$: $2 \le R < 3$;
- $S_j \in [4, 6]$: $1 \le R < 2$;
- $S_j \in [0, 3]$: $R < 1$.

Completeness (S_m). Coverage is measured by

$$M = \frac{N_{\text{addressed requirements}}}{N_{\text{total requirements}}}.$$

The rubric is:

- $S_m = 10$: $M = 1$;
- $S_m \in [7, 9]$: $M \ge 0.8$;
- $S_m \in [4, 6]$: $0.5 \le M < 0.8$;
- $S_m \in [0, 3]$: $M < 0.5$.

Physical consistency (S_p). Physical validity is assessed via dimensional analysis, unit propagation, and limiting-case behavior:

- $S_p = 10$: dimensionally consistent, correct limits, physically plausible;
- $S_p \in [7, 9]$: generally consistent with minor issues;
- $S_p \in [4, 6]$: some unit or limiting-behavior errors;
- $S_p \in [0, 3]$: unphysical or dimensionally inconsistent results.

Critic score. When Critic feedback is included, it is scored separately:

$$S_{\text{critic}} = C_a + C_d + C_f + C_t + C_c + C_h + C_v + C_o,$$
$$S_{\text{critic}} \in [0, 100]$$

with:

$C_a \in [0, 20]$	accuracy of identification,
$C_d \in [0, 15]$	depth of analysis,
$C_f \in [0, 15]$	constructive feedback quality,
$C_t \in [0, 15]$	technical understanding,
$C_c \in [0, 10]$	clarity of communication,
$C_h \in [0, 10]$	comprehensiveness,
$C_v \in [0, 10]$	pedagogical value,
$C_o \in [0, 5]$	objectivity and fairness.

This score is recorded but does not feed the pipeline decisions or the present analysis. In the dynamical language of Appendix E, it could become a second observable alongside the Actor score, separating poor feedback from useful feedback that the Actor fails to take up. We leave this two-observable analysis for follow-up work.

Critic accuracy (C_a). Error-detection quality is quantified using precision and recall,

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}.$$

High scores require both high precision and high recall.

Critic depth (C_d). Depth of analysis is measured by

$$D = \frac{N_{\text{examined steps}}}{N_{\text{total steps}}}.$$

Constructiveness (C_f). Actionability is measured by

$$A = \frac{N_{\text{actionable suggestions}}}{N_{\text{total suggestions}}}.$$

Technical understanding (C_t). Conceptual accuracy is measured by

$$T = \frac{N_{\text{correct concepts}}}{N_{\text{referenced concepts}}}.$$

Comprehensiveness (C_h). Coverage of relevant issues is measured by

$$H = \frac{N_{\text{addressed aspects}}}{N_{\text{problem aspects}}}.$$

Pass criterion. A single Actor turn is marked as passing only if all of the following hold:

$$\text{Pass} = \Big(\text{Eq}(A, R) = 1\Big) \wedge \Big(S_{\text{actor}} \geq 80\Big) \wedge \Big(S_c \geq 40\Big).$$

In implementation, a non-equivalent final result always forces $\text{Pass} = 0$. A run is then counted as converged if at least one of its iterations satisfies this turn-level pass rule under the scoring Judge (Section 2.2).

Reference-free use. For known textbook problems, correctness and convergence rely on the reference answer through $\text{Eq}(A, R)$. In a future open-problem setting this part of the rubric would have to be replaced by external validation: symbolic checks, limiting cases, simulations, experiments, theorem-prover output, or expert consistency review. The process criteria — mathematical rigor, logical flow, justification quality, completeness relative to the stated task, and physical consistency — remain meaningful without a reference solution. By contrast, "correctness", Critic error-identification accuracy, and the present pass rule are reference-dependent and would become measures of constraint satisfaction rather than answer matching. In the dynamical-systems language of Appendix E, removing the reference solution would keep the score projection but alter the drift, since the Critic's directional pull is currently reference-conditioned.

Error taxonomy. The Judge also records categorical errors. At the solution level, major issues are classified into:

`COMPUTATIONAL_ERROR`, `CONCEPTUAL_ERROR`, `METHODOLOGICAL_ERROR`, `DIMENSIONAL_ERROR`, `BOUNDARY_CONDITION_ERROR`, `CONVERGENCE_ERROR`, and `APPROXIMATION_ERROR`.

Most categories are self-explanatory. The less obvious labels are: `METHODOLOGICAL_ERROR`, for using an inappropriate setup or solution method; `BOUNDARY_CONDITION_ERROR`, for missing or misapplied boundary/normalization conditions; `CONVERGENCE_ERROR`, for invalid limiting, integral, or series-convergence reasoning; and `APPROXIMATION_ERROR`, for unjustified expansions or dropped terms.

Binary error flags. The Judge records the following binary diagnostic flags:

`SIGN_ERROR`, `MISSING_TERM`, `PRODUCT_RULE_ERROR`, `CHAIN_RULE_ERROR`, `BOUNDARY_CONDITION_MISAPPLIED`, `UNIT_MISMATCH`, `ALGEBRA_SIMPLIFY_FAIL`, `LIMIT_ERROR`, `NORMALIZATION_ERROR`, `INCOMPLETE_EXPR`, `TEST_FAIL`, `TENSOR_INDEX_ERROR`, `DIMENSIONAL_CONSISTENCY`, `SYMMETRY_PRESERVATION`, `COMBINATORIAL_FACTORS`, `REGULARIZATIONAL_CONSISTENCY`, `MISSING_JUSTIFICATION_NON_TRIVIAL_STEPS`, `LIMITING_BEHAVIOR_FAIL`, `POSITIVITY_AND_REALITY_CONSTRAINTS_VIOLATION`, `VIOLATION_OF_WARD_IDENTITIES`.

Most labels are self-explanatory. The less standard ones correspond to failed problem-specific checks: `TEST_FAIL` marks failure of an explicit validation test, `COMBINATORIAL_FACTORS` covers missing symmetry or counting factors, and `VIOLATION_OF_WARD_IDENTITIES` marks failure of gauge/current-conservation constraints. The names are legacy JSON keys, so some positive-sounding labels are interpreted as diagnostic indicators when set to true.

Missing-content labels. The Judge may also assign missing-content labels to the Actor solution:

`UNJUSTIFIED_CLAIMS`, `INCORRECT_RESULT`, `INCOMPLETE_DERIVATION`, `MISSING_BOUNDARY_CONDITIONS`, `DIMENSIONAL_INCONSISTENCY`, and `CONVERGENCE_FAILURE`.

Progress signal. To assess whether Critic intervention changed the solution appreciably, the Judge also records an Actor–Critic progress indicator:

$$\text{Progress} = \mathbf{1}(|S_{\text{current}} - S_{\text{previous}}| > \delta)\,,$$

where δ is a threshold, typically set to 5 points.

Implementation note. The Judge is required to return a strict JSON object containing the above fields. If the returned structure is invalid or unparsable, the system defaults to a failing verdict and records the malformed response for debugging.

The component-score diagnostic in Figure 8 decomposes the Actor score into the six rubric criteria defined in Section 2.2. Each line is normalized to the maximum value of that criterion, so the plot shows which parts of the Judge score move during the dialogue rather than how many raw rubric points each criterion contributes. This is not an additional aggregate evaluation metric used in the statistical tests; it is a decomposition of the score $\bar{s}$ summarizes. The figure is descriptive: because SCALAR stops runs after a passing verdict, later turns average over the subset of runs still active at that turn.

D. Evaluation Metrics and Statistical Procedures

This appendix supplements Section 2.2 with details not in the main text: the precise pass rule used in the convergence count, edge cases for runs that terminate early, and the specific statistical test implementations. The scoring rubric itself (the six Judge dimensions, pass criterion, and error flags) is described in Appendix C.

Iteration range and early stopping. The iteration cap is $T_{\max} = 4$ for the DeepSeek-family runs and $T_{\max} = 5$ for the Haiku runs; a run terminates earlier at $T < T_{\max}$ when the Judge used during generation issues a passing verdict or when the score-stagnation trigger fires. The reported evaluation metrics use the actual T of each run, so $\bar{s}_i$ and g_i are comparable across runs of different length. Runs that terminate after a single iteration (T=1) have g_i=0 by construction.

Pass rule behind the convergence rate. The turn-level pass rule from Appendix C requires final-answer equivalence with the reference, total Actor score ≥ 80, and correctness component ≥ 40; a non-equivalent final answer forces Pass=0 regardless of the other components. A run is counted as converged if any iteration of the dialogue satisfies this rule under the scoring Judge. For runs whose original loop was driven by the same Judge that scores them this is equivalent to a final-turn pass; for re-scored transcripts the loop length is fixed by the original Judge and the scoring Judge can mark a non-terminal iteration as the convergence event.

Statistical tests. Each run contributes one independent observation for each response variable. All tests are two-sided, non-parametric (no Gaussianity assumption on the score distributions), and use the default `scipy.stats` implementations, with the classical tests cited below for provenance.

Unless explicitly stated otherwise, the statistical response variables are the raw run-level evaluation quantities $\bar{s}_i$, g_i, and r_i. The problem-normalized contrasts $D_{\bar{s}}$ and D_R are descriptive summaries used in Figure 5; they are not inputs to the p-values reported in the Results.

Throughout the paper we use the conventional $p < 0.05$ threshold to flag "statistically significant" results: this means the test statistic is at least as extreme as the 95th percentile of its null distribution — equivalently, under the null hypothesis (no real effect) a result this surprising or more would occur less than 1 in 20 times; smaller p-values carry stronger

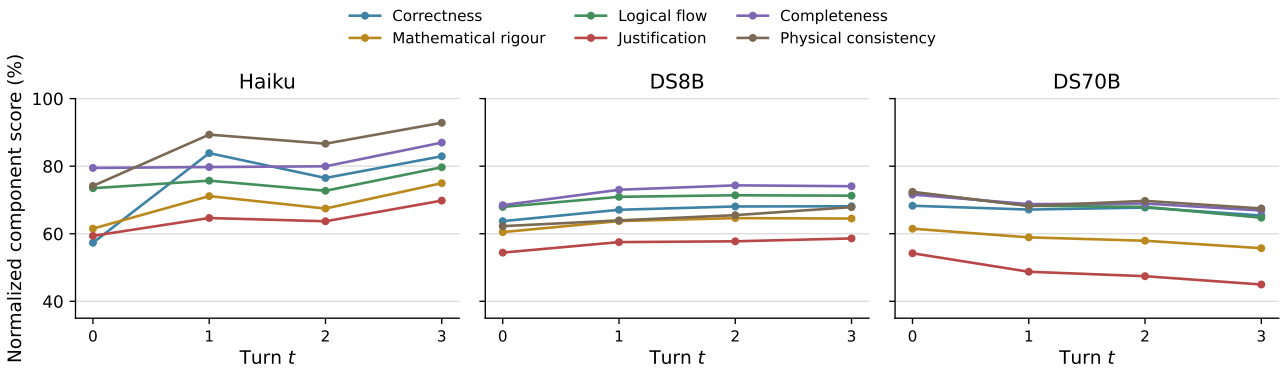

Figure 8. Actor score-component evolution under QWQ scoring. Each component is normalized to its rubric maximum before averaging. Haiku uses the two Peskin problems; DS8B and DS70B use all three problems. This diagnostic decomposes the Judge score and is not an additional aggregate evaluation metric.

evidence against the null. In the main text, each quoted p-value belongs to the statistical test named in the same sentence, usually a Kruskal–Wallis omnibus test across Critic strategies or Actor personas. When a re-scoring check is mentioned without a new p-value, it is used only as a sensitivity check on the direction of the conclusion, not as a separate headline test.

- **Wilcoxon signed-rank** on paired (s_0, s_{T-1}) scores (`scipy.stats.wilcoxon`; Wilcoxon, 1945). A paired-sample test that asks whether the median of the within-run gains $g_i = s_{i,T_i-1} - s_{i,0}$ differs from zero; we use it to assess whether the multi-turn dialogue induces a non-zero gain over the population of runs.

- **Kruskal–Wallis H-test** on $\bar{s}$ across the five Critic strategies (`scipy.stats.kruskal`; Kruskal & Wallis, 1952). A rank-based multi-group test that asks whether samples from ≥ 2 groups come from a common distribution; we use it to assess whether the Critic feedback strategy label has any overall effect on $\bar{s}$, applied separately for each Actor–Critic pairing.

- **Mann–Whitney U-test** for specific pairwise Critic feedback strategy comparisons (`scipy.stats.mannwhitneyu`; Mann & Whitney, 1947). A two-sample rank-based test that asks whether values from one group tend to be larger than values from another; we use it to assess whether a particular pair of Critic feedback strategies (*e.g.*, pedagogical vs. strict) differ. We report uncorrected p-values; where multiple comparisons contribute to the same claim we flag this alongside the relevant result. We do not use rank correlations between five Critic feedback strategy means as evidential tests, because with only five ranks they are too coarse to support an independent statistical claim.

Re-scoring protocol. Re-scoring Judges score the stored Actor–Critic transcripts using the same Judge prompt (Appendix A) but a different Judge LLM; the Actor and Critic are not re-run, and the original stopping decisions are preserved. The Haiku transcripts are re-scored by QWQ and by DS70B in addition to the primary Sonnet scoring, and the DS70B and DS8B transcripts are re-scored by DS70B in addition to the primary QWQ scoring. This supports the cross-Judge consistency checks reported in Section 3.

E. Score-Update Field and Markov Projection

The score-update curves in Figure 4 estimate a one-step score drift; they are not inputs to the hypothesis tests in the Results. Let X_t denote the full SCALAR state after Actor turn t: problem, persona, Critic feedback strategy, transcript, scoring state, and model-sampling rule. Conditional on the experimental cell, the algorithm induces a one-step transition

$$p(X_{t+1} \mid X_t, X_{t-1}, \ldots) = p(X_{t+1} \mid X_t).$$

The scalar Judge score is a projection $s_t = S(X_t)$, and the useful question is how much of the dialogue dynamics remains visible in this one-dimensional coordinate. For run i, let $s_{i,t}$ be the Judge score after Actor turn t and define $\Delta s_{i,t} = s_{i,t+1} - s_{i,t}$. For a score bin B_b, we estimate the empirical drift

$$\hat{v}_b = \frac{1}{|\mathcal{T}_b|} \sum_{(i,t)\in\mathcal{T}_b} \Delta s_{i,t}, \qquad \mathcal{T}_b = \{(i,t) : s_{i,t} \in B_b\}.$$

The shaded bands in Figure 4 are bootstrap confidence intervals for $\hat{v}_b$ over runs. When the fitted curve crosses zero, we denote the crossing by s^* and call it a projected fixed point. This means only that the observed next-turn update vanishes on average near that score; it is not necessarily a complete fixed point of the full recorded dialogue state.

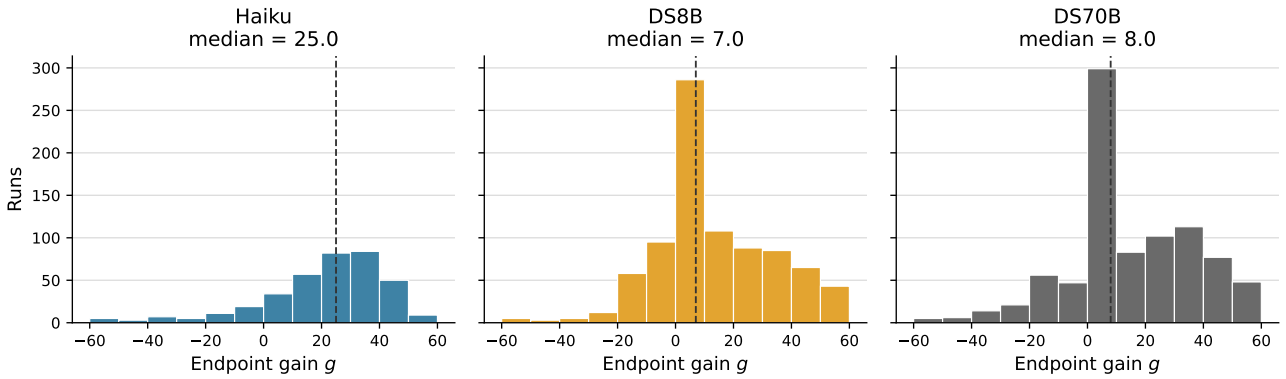

Figure 9. Endpoint gain distributions under QWQ scoring. Haiku uses the two Peskin problems (n=383), while DS8B and DS70B use all three problems (n=900 each). Dashed vertical lines mark the median gain.

One can also estimate the local fluctuation scale

$$\hat{D}_b = \frac{1}{2}\,\mathrm{Var}_{(i,t)\in\mathcal{T}_b}(\Delta s_{i,t})\,,$$

the empirical analogue of diffusion for the projected score. Here stochasticity comes both from nonzero-temperature LLM sampling and from projecting many transcript states to the same score. We do not fit a continuous Fokker–Planck or Smoluchowski model here: the trajectories are short, early stopping censors the active ensemble, and passing depends on final-answer equivalence rather than on a score threshold alone. A future temperature sweep could separate drift-dominated improvement from noise-assisted escape out of low-drift regions.

Discarding the transcript gives a score-only Markov closure,

$$p(s_{t+1} \mid s_t, s_{t-1}, \ldots) \approx p(s_{t+1} \mid s_t),$$

after conditioning on the experimental cell. This does not follow from the transcript-level Markov property: two transcripts with the same score can contain different physics errors. As an exploratory DS70B/QWQ check, we regressed the residual $\Delta s_t - \hat{v}(s_t)$ on the previous score s_{t-1} within each problem; the fitted memory terms were small compared with their standard errors. Thus the present data are consistent with the score-only closure, while longer held-out trajectories are needed for predictive validation.

The predictive version would fit a discrete transition kernel over score bins with an absorbing pass state $\mathcal{P} = \{s \geq 80 \text{ and final answer equivalent}\}$. On held-out trajectories this kernel could predict score histograms, convergence rates, or mean remaining turns; in the absorbing-chain idealization, the non-pass bins obey $\tau = (I - K_{BB})^{-1}\mathbf{1}$.