Reasoning LLMs for Materials Discovery with Physics-aware Rejection Sampling

Lee Hyun[1] **Sohee Yoon**[1] **Jinwoo Park**[1] **Sue In Chae**[1] **Seongeon Park**[1]

Jooyeon Ahn[2] **Yebin Jung**[2] **Youjung Chung**[2] **Hogeun Chang**[2] **Myeonginn Kang**[1]

Jina Kim[2] **Sujin Park**[2] **Ho-Gyeong Kim**[1] **Myeonghun Jeong**[1]*

[1] Materials AI Lab (AI Center), Samsung Electronics
[2] Samsung Advanced Institute of Technology (SAIT)
{hyun103.lee, mh29.jeong}@samsung.com

Abstract

AI-driven materials discovery that couples automated experimentation with algorithmic decision-making requires process aware recipe to property predictors that are accurate, calibrated, and physically admissible. We approach this as a reasoning problem with large reasoning models (LRMs). To instill reasoning capability into language models, we curate reasoning traces from a teacher model to train a student model. However, most training pipelines select reasoning traces using binary correctness or learned preference signals that poorly reflect physical admissibility. We introduce *Physics-aware Rejection Sampling (PaRS)*, a training-time trace selection scheme that favors traces consistent with fundamental physics and numerically close to targets, with lightweight halting to control compute. We instantiate our framework with a large student model fine-tuned on traces synthesized by a larger teacher model, and evaluate under matched token budgets against various rejection sampling baselines. Our method improves accuracy and calibration, reduces physics-violation rates, and lowers sampling cost relative to baselines. These results indicate that modest, domain-aware constraints combined with trace-level selection provide a practical path toward reliable, efficient LRMs for process-aware property prediction and closed-loop materials design.

1 Introduction

A central goal in materials discovery is to compress the experimental loop by coupling automated experimentation with algorithmic decision making. Within this loop, property prediction is the core module. Accurate models that map composition, structure, and process/recipe variables to target properties (e.g., materials properties or device-level figures of merit) convert combinatorial exploration into tractable optimization and enable closed-loop design via bayesian optimization, provided that the models expose calibrated uncertainty and lightweight physics constraints to keep proposals physically admissible [1, 2, 3, 4, 5, 6]. Prior work has largely targeted properties from composition or crystal structure [7, 8] and, more recently, has explored text- or instruction-conditioned surrogates with LLMs [9]; yet scaling these predictors to process-aware, recipe to property tasks at the device level remains challenging [10, 11, 12].

*Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Science.

Recently, large reasoning models (LRMs)—language models trained and/or reinforced to produce reliable reasoning traces—have shown dominant performance in diverse areas such as math, coding and scientific QA [13, 14, 15]. Their step-wise reasoning capabilities are a natural fit for recipe to property prediction, where multi-step physical arguments along the classic *chemical composition* $\rightarrow$ *process* $\rightarrow$ *micro-structure* $\rightarrow$ *property* chain are essential and well established in integrated computational materials engineering [16]. Despite the promise, the leveraging LRMs on property prediction task remains underexplored relative to LLM and naive knowledge extraction [17, 9]. In this paper, we study how to train LRMs that reason effectively about materials recipes and output numerically correct, physically grounded properties.

A prevailing strategy for training LRMs to reason is to use filtered or re-weighted training signals based on the quality of teacher generated traces, complemented by test-time scaling [18]. Concretely, models are fine-tuned on self-generated rationales kept only when they reach correct outcomes [19, 20], or on samples ranked by a learned reward/verifier [21, 22, 23], or they aggregate multiple samples at decoding [24]. These methods work as standard building blocks for LRMs along with SFT-only pipelines and RL-style post-training [25, 26].

We argue that training property prediction LRMs from generated reasoning traces requires more sophisticated, physcially grounded rejection sampling. Two characteristics of this task drive the need: *(1) High combinatorial design space.* The composition, process, structure, property chain creates high-dimensional, multi-mechanism spaces; inverse maps are often non-unique, yielding traces that seem plausible yet are scientifically incorrect [27, 28, 29, 30, 31]. Therefore, effective learning requires sufficient exploration that searches both the design and reasoning trace space. *(2) Physically grounded outputs.* Targets are physical quantities whose magnitudes are constrained by physics and even small numeric deviations matter. Filters must therefore enforce admissible ranges and physical constraints from conservation laws and constitutive relations rather than rely solely on binary correctness.

Motivated by these challenges, we propose *Physics-aware Rejection Sampling (PaRS)*, a domain-tailored approach to optimize reasoning traces. Unlike prior methods that depend on binary correctness or learned reward models, our method couples rejection sampling with task-native, continuous error metrics derived from wet-lab experiments. Concretely, for each device recipe, we sequentially generate candidate traces, accepting the first trace that satisfies physics-aware acceptance gates and halting sampling early when further candidates show negligible variance or improvement.

We adopt Qwen3-32B [2] as the backbone model, fine-tuned via supervised fine-tuning (SFT) on internal prompts, with teacher reasoning traces synthesized from Qwen3-235B [3] [15]. We benchmark against various rejection sampling methods under matched token budgets. Empirically, our method achieves the highest overall accuracy and calibration, while also delivering superior compute efficiency compared to existing baselines.

Our contributions are threefold.

- We formulate recipe to property prediction as a reasoning task where physics-aware verification is essential.
- We propose novel physically grounded rejection sampling for optimizing reasoning traces, introducing the combination of powerful gating and halting techniques.
- We conduct (1) a teacher-side ablation, comparing our physics-aware sampler against six baselines in terms of trace accuracy and sampling efficiency, and (2) a student-side evaluation, fine-tuning an open-source LRM to demonstrate consistent gains in accuracy, calibration, and compute efficiency over all baselines.

2 Related Work

LLMs for materials design. Recent advances in large language models (LLMs) have demonstrated strong generalization capabilities in materials design, drawing on interdisciplinary knowledge from chemistry, physics, and engineering [32, 33]. Beyond general-purpose LLMs, domain-specific models such as MatSciBERT [34], MatBERT [35], and MELT [36] have been trained on large-scale materials

[2]https://huggingface.co/Qwen/Qwen3-32B

[3]https://huggingface.co/Qwen/Qwen3-235B-A22B

science corpora, successfully capturing fundamental concepts that link structure, properties, processes, and performance. A key step in the materials design pipeline is accurate property prediction from symbolic representations, which serves as a surrogate for expensive experiments and enables rapid candidate screening. Leveraging their ability to process unstructured scientific data, recent studies have applied LLMs to this task without the need for elaborate feature engineering. For example, LLM-Prop [37] employs LLMs to predict crystalline material properties, while Li et al.[38] integrate LLMs with graph neural networks for improved prediction accuracy. In the context of quantum-dot materials, Choi et al.[39] developed LLM-based synthesis protocol generation and property prediction models, fine-tuned on proprietary synthesis datasets. LLMs have also been explored as surrogate models in optimization frameworks. LLAMBO [40] utilizes the exploration capability of LLMs within Bayesian optimization, and BOPRO [41] incorporates LLM-based search strategies that exploit evolving uncertainty estimates to propose promising candidates in each iteration, thereby accelerating the discovery of globally optimal solutions.

Large Reasoning Models (LRMs). The paradigm of next-token prediction has undergone a significant shift with the introduction of "thought" concept—a sequence of intermediate steps representing a model's internal reasoning process [14, 13, 15, 18]. This innovative approach enables LLMs to mimic complex human reasoning, such as reflective thinking and tree search. Chain-of-Thought (CoT) prompting [42] initially demonstrated that few-shot rationales could unlock sophisticated reasoning in LLMs, while Self-Consistency [43] further enhanced reliability by marginalizing over multiple reasoning paths. Tree-of-Thoughts (ToT) [44] later reframed inference as a search over partial thought sequences, incorporating look-ahead and backtracking. Beyond prompting, process supervision has emerged as a powerful technique, training step-level verifiers or reward models to guide the reasoning process. This approach has been shown to outperform models trained with outcome-only labels, particularly in mathematical reasoning tasks [45]. More recently, GPT-o1 [14], Qwen3 [15], and DeepSeek-R1 [13] have popularized modern Large Reasoning Models (LRMs) by integrating long-form thinking with process supervision, RL-based post-training, and test-time scaling. Building on these foundational advances, our work adapts these reasoning mechanisms to the domain of materials design, specifically targeting physically grounded recipe to property prediction.

Rejection sampling for LLMs. Rejection sampling is widely recognized as an effective data-filtering method that promotes higher-quality supervision. In the context of RLHF and preference-optimization, it narrows multiple generated outputs per prompt to only the high quality responses, as determined by a reward model during post-training adjustments. For example, RAFT[46] aligns generative models efficiently by using a reward model and abundant candidate samples; it discards outputs demonstrating undesired behaviors and fine-tunes the model solely on the selected high-quality subset. Building on this, Reinforce-Rej[47] proposes a minimalist policy-gradient extension that filters out both entirely incorrect and entirely correct samples, enhancing stability and efficiency. In reasoning tasks, STAR-like models[48, 49, 50] eliminate expensive human annotations by employing a self-taught reasoning loop—generating chain-of-thought traces, self-verifying correctness, and fine-tuning only on reliable examples. Additionally, Rejection Sampling Fine-Tuning (RFT)[51] enhances mathematical reasoning by incorporating model-generated reasoning traces filtered for correctness into training.

3 Method

3.1 Task Definition

a Quantum dot light-emitting diodes (QD-LEDs) are electroluminescent devices that use colloidal semiconductor quantum dots as the emissive layer, offering narrowband spectra and composition tunable color [52, 53]. In practice, closed-loop materials design must optimize figures of merit measured at the device level—e.g., peak external quantum efficiency and operational stability—because these metrics ultimately determine application viability for emissive displays and lighting (e.g., televisions, monitors) [54, 55, 53]. We therefore frame inputs as complete device recipes: a multi-layer stack typically consist of anode/ITO, hole-injection layer (HIL), hole-transport layer (HTL), quantum-dot emitting layer (EML), electron-transport layer (ETL), electron-injection layer/cathode with per layer materials and process parameters. Each layer records identifiers (material, formulation), geometric parameters (e.g., thickness), and process variables (e.g., solution concentration, spin profile, bake/an-

```
QD-LED recipe:
  substrate:
    type: ITO/glass, thickness_nm: 150, rsheet_ohm_sq: 15, roughness_Rq_nm: 1.5
    pretreat: UV-ozone, 10 min; solvent rinse (IPA); 120 C annealing 10 min
  stack:
    [HIL layer]
      substances: PEDOT:PSS (AI4083), thickness_nm: 40, filtration_um: 0.45
      work_function_eV: 5.1, process: spin (4000 rpm, 60 s -> 8000 rpm, 5 s ramp)
      annealing (150 C, 10 min, air)
    [HTL layer]
      substances: Poly-TPD, thickness_nm: 20, HOMO_eV: -5.2
      solution: 8 mg/mL, chlorobenzene (99.8%, anhydrous), filtration_um: 0.2
      process: spin(3000 rpm, 45 s); annealing(120 C, 10 min, N2)
    [EML layer]
      substances: CdSe/ZnS core/shell QDs (green), emission_peak_nm: 525,
      FWHM_nm: 22, core_diameter_nm: 5.5, ligand_primary: oleic acid / oleylamine,
      solution_conc_mg_mL: 25, solvent: octane (anhydrous, <10 ppm H2O)
      filtration_um: 0.2 PTFE, target_areal_density_ug_cm2: 40, thickness_nm: 25,
      process: spin (2000 rpm, 30 s); annealing (80 C, 5 min), film_roughness: 1.8
      PLQY_solution_fraction: 0.92, PLQY_film_fraction: 0.80
    [ETL layer]
      substances: ZnO nanoparticles (sol-gel/colloidal), mean_particle_diam_nm: 5
      thickness_nm: 30, solution: 10 mg/mL, isopropanol
      process: spin(3000 rpm, 30 s); annealing (90 C, 5 min, N2)
        ...
```

Figure 1: Structured QD-LED recipe example.

```
You are a world-class expert in quantum-dot light-emitting-diode (QD-LED) device
physics and fabrication.

<Query QD-LED recipe>

TASK: Predict external quantum efficiency for a QD-LED device fabricated by the
query recipe.

Final output format (only json output)
Please provide your final report in a structured JSON format.
{
  "answer": <PREDICTED_VALUE> %
}
```

Figure 2: `Prompt` for the property prediction task with large reasoning models (LRMs).

neal temperature and duration, atmosphere), along with post-process steps (e.g., UV–ozone, plasma, solvent rinse). See Figure 1 for a recipe example.

We formulate QD-LED device property prediction as a reasoning LLM task. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where x_i is a QD-LED recipe as above and $y_i \in \mathbb{R}$ is a device-level target; in this work we focus on max external quantum efficiency (i.e., $y_i = \mathrm{EQE}_{\max}$). Given a task-specific `prompt` (See Figure 2) and recipe x_i, a large reasoning model f_θ produces a reasoning trace τ along with a numeric prediction $\hat{y}$:

$$f_\theta(\texttt{prompt}, x_i) \to (\tau, \hat{y}). \tag{1}$$

3.2 PaRS: Physics-aware Rejection Sampling

For training LRMs, the supervision signal extends beyond the final answer to include the sampled reasoning traces themselves. Achieving high-quality supervision for reasoning LLMs therefore requires rejection sampling to filter out suboptimal traces. As illustrated in Figure 3, we propose *Physics-*

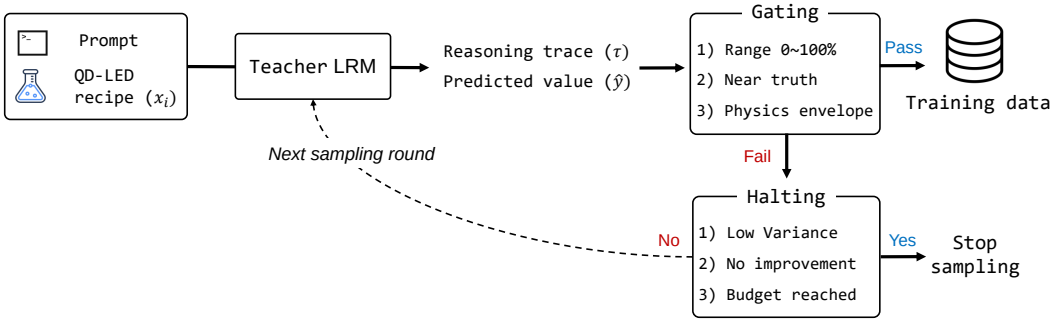

Figure 3: PaRS workflow: the teacher generates a mini-batch of candidates; a candidate is accepted only if it passes gates (range, near-truth tolerance, physics envelope). If none pass, halting checks decide whether to stop or raise temperature and continue to next sampling round. Accepted traces supervise the student model as training data.

aware Rejection Sampling (PaRS), which integrates physics-aware gating and halting mechanisms to optimize reasoning traces

Physics-aware Gating. For each recipe x_i, we generate reasoning traces sequentially from the teacher model up to K. We accept the first sample that satisfies all of following acceptance gates:

$$\hat{y}^{(k)} \in [0, 100], \tag{2}$$

$$\left|\hat{y}^{(k)} - y_i\right| \leq \varepsilon_{\mathrm{MAE}}, \tag{3}$$

As $\mathrm{EQE}_{\max}$ is reported as a percentage, Eq. (2) enforces range consistency. Unlike categorical correctness filters [19, 20, 21, 24, 18], Eq. (3) uses a continuous error against wet-lab ground truth, yielding richer learning signals.

With external quantum efficiency (EQE) as the prediction target, we can define an empirical upper bound $U(x)$ as follows. EQE is commonly factorized as $\mathrm{EQE} = \eta_{\mathrm{out}} \cdot \eta_{\mathrm{rad}} \cdot \gamma$ with $\eta_{\mathrm{out}} \leq 1$ and $\gamma \leq 1$ by definition. In regimes where solid-state PLQY limits the radiative yield, the conservative relation $\eta_{\mathrm{rad}} \leq \mathrm{PLQY}$ holds, implying $\mathrm{EQE} \leq \mathrm{PLQY}$ [56, 57, 58]. PLQY (photoluminescence quantum yield) denotes the ratio of emitted to absorbed photons; we use the solid-state PLQY of the EML under device-relevant conditions. We therefore define an empirical, recipe-specific upper bound as $U(x)$=$U_{\mathrm{PL}}(x)$, where $U_{\mathrm{PL}}(x)$ is instantiated by the highest measured film photoluminescence quantum yield of the emissive layer in the recipe x. We add this upper bound to the acceptance gates:

$$\hat{y}^{(k)} \leq U(x_i) \tag{4}$$

It prevents reasoning traces that predict physically implausible overshoots for target property. If no sampled candidate satisfies Eq. (2)–(4), we discard the example.

Adaptive Halting. We sample in mini-batches of size b and proceed round by round until the total budget $K_{\max}$ is exhausted. In round r ($r = 1, 2, \ldots$), we draw exactly b candidates $\{(\tau^{(r,j)}, \hat{y}^{(r,j)})\}_{j=1}^{b}$ at temperature T_r. For each candidate, we apply the acceptance gates in Eqs. (2)–(4). If any candidate passes, we accept the earliest passing one and terminate sampling.

If no candidate in the mini-batch passes, we apply two halting checks before proceeding. (i) *Variance-based halting* (from round 1): stop when the within-batch error variance falls below a threshold, indicating insufficient diversity to justify further exploration. (ii) *Improvement-based halting* (from round 2): stop when the best error in the current round fails to improve over the previous round by at least a small margin. We also halt once the cumulative number of sampled candidates reaches $K_{\max}$. If none of these conditions trigger, we increase T_r and continue to the next round to encourage exploration. Refer Appendix A.1 for the details of halting methods.

Table 1: Analysis of reasoning trace selection. All traces are generated with QWEN3-235B. Higher LLM-as-a-Judge score and lower MAE is better. K_{avg} is the average number of generated reasoning traces per prompt. Our Halting logic yields K_{avg}=6.4 on average (fewer tokens/calls per accepted trace). 'Budget→selected" counts teacher generates per prompt and the number retained after selection.

Method	K_{avg}	**Budget (→ selected)**	**LLM-as-a-Judge (score, 0–10)**	**MAE**
No sampling	1.0	$1 \rightarrow 1$	5.97	2.440
Random sampling	12.0	$12 \rightarrow 1$	5.86	2.327
Longest rationale selection	12.0	$12 \rightarrow 1$	6.10	2.274
Self-consistency selection	12.0	$12 \rightarrow 1$	5.78	<u>1.829</u>
LLM-as-a-Judge	12.0	$12 \rightarrow 1$	<u>6.55</u>	2.223
Multi-sampling	12.0	$12 \rightarrow 12$	5.89	2.356
PaRS (ours)	**6.4**	$12 \rightarrow 0.8^{\dagger}$	**7.51**	**0.829**

† We drop the around 20% of sample that not passed our acceptance logic in Sec. 3.2 thus 0.8 traces kept per prompt on average.

4 Experiments

4.1 Baselines

We curate 11k QD-LED device dataset and split into 10k for training and 1k for testing. We construct prompts from all 11k dataset for a property prediction task and query 10k train prompts with QWEN3-235B to sample teacher reasoning traces. To ensure fair comparison, each reasoning trace has a same sampling budget of K=12, except for no sampling. We compare the following methods for selecting reasoning traces including our method.

1. **No sampling:** use the first generated trace.
2. **Random sampling:** uniformly sample one of the K traces.
3. **Self-consistency aggregation:** select the trace whose final answer is closest to the median across all K answers [24].
4. **Longest trace:** select the trace with the largest token length.
5. **LLM-as-a-judge:** score all traces with a larger judge model (DeepSeek-R1 [4]) and select the top-ranked trace [59]. Refer Appendix. A.2 for details of the LLM-as-a-Judge prompting.
6. **Multi-sampling:** retain *all* K traces as supervision [60].
7. **PaRS (ours):** mini-batch size b=4 with a temperature schedule $T \in \{0.6, 0.8, 1.0\}$ increasing by 0.2 per round. We set ε_{MAE}=1, ε_{var}=1 and δ_{imp}=1 by analyzing the data distribution. If no candidate passes the gates within the budget, the example is discarded.

After rejection sampling, we fine-tune QWEN3-32B as the student model for a single epoch on the traces selected by each method, using AdamW (learning rate 2×10^{-5}) on 32×A100 (80 GB) GPUs; unless stated otherwise, all other training hyperparameters are shared across methods.

Evaluation metrics. We report two groups of metrics. (1) Teacher-side trace quality: mean absolute error of the selected trace' prediction, and the average number of sampled traces per prompt to quantify the cost of constructing the selected traces. To further assess trace quality, we also employ a larger LLM (DeepSeek-R1) as an external judge, providing external evaluation of reasoning quality on a 0–10 scale. (2) Student-side performance: MAE, R^2, Spearman's ρ, and a physics violation rate—the fraction of predictions that fall outside $[0, 100]$ or exceed the empirical upper bound $U(x)$ on the hold out test set.

For each test prompt, we run five independent inferences with the trained student models and take the median of the five predictions before computing MAE, R^2, and Spearman's ρ. For the physics

[4]https://huggingface.co/deepseek-ai/DeepSeek-R1-0528

Table 2: QD-LED property prediction with QWEN3-32B trained on teacher traces selected by each method from QWEN3-235B. Lower MAE is better; higher R^2 and Spearman ρ are better. Viol.% is computed on test predictions without post hoc clipping to $U(x)$. # train is the number of supervision traces used for SFT: multi-sampling retains all K=12 traces per example (thus 12× larger), whereas our method keeps only candidates that pass our gates, resulting in fewer traces.

Training data (prompt + reasoning trace)	# train	MAE (pp)	R^2	Spearman ρ	Viol. (%)
no rejection sampling	10 000	2.001	0.376	0.607	35.8
random sampling	10 000	1.961	0.358	0.621	35.2
longest sampling	10 000	1.942	0.375	0.614	35.3
self-consistency sampling	10 000	1.933	0.377	0.629	36.8
LLM-as-a-Judge	10 000	1.889	0.408	0.667	35.4
multi-sampling	120 000	1.984	0.335	0.632	36.6
PaRS (ours)	8000	**1.808**	**0.424**	**0.705**	**27.7**

violation rate, we evaluate the constraint indicator on each of the five predictions and report the average violation frequency across all ensemble. Details of the LLM-as-a-Judge prompting procedure are provided in Appendix A.2.

4.2 Results

Our experiments show that PaRS effectively optimize the teacher' reasoning traces to induce reasoning capability for property prediction on student model. We present our findings in two parts: first, an analysis of the trace selection process itself, and second, an evaluation of the trained student LRMs on property prediction task.

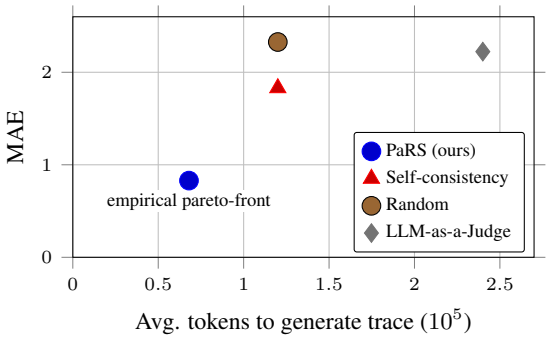

Figure 4: Compute–accuracy frontier for rejection sampling methods. Our approach achieves the lowest teacher MAE with substantially fewer required tokens, forming the empirical Pareto front. The x-axis shows average required tokens for generating reasoning trace per prompt and the y-axis shows teacher MAE. See Appendix A.3 for details.

Analysis of reasoning trace sampling. We assess rejection sampling strategies on traces generated by the teacher QWEN3-235B. As summarized in Table 1, PaRS achieves lower prediction error while requiring fewer generations on average, placing it on the empirical quality–efficiency Pareto front (Fig. 4). A simple MAE gate could make low error appear trivial, yet an external judge that is not optimized for our metric still assigns PaRS the highest overall score. Self-consistency, which often lowers MAE, receives the weakest judge score among baselines. This gap suggests that generic preference signals do not fully align with physics-grounded proximity to ground truth. By combining physics-aware gates with temperature scheduling and early stopping, PaRS concentrates supervision on high-fidelity, physically admissible traces rather than on merely "good-looking" rationales.

Evaluation of trained LRMs. Training the student QWEN3-32B on traces selected by our method yields consistent gains in accuracy, correlation with ground truth, and physical admissibility, while using substantially fewer supervision traces than competing approaches (See Table 2). The LLM-as-a-Judge baseline is competitive in error and correlation but does not match the reduction in violations or the calibration gains achieved by PaRS. Retaining all traces increases supervision volume yet underperforms, consistent with amplified label and reasoning noise when traces remain unfiltered.

These improvements arise from aligning the acceptance rubric with a continuous, physically grounded target. The range check in Eq. (2) and the empirical envelope in Eq. (4) for target EQE suppress

implausible overshoots. The continuous gate in Eq. (3) rewards numerical proximity to wet-lab measurements rather than binary correctness. Since the mapping from recipe to property is many to one and complex, enforcing zero error steers supervision toward outliers and collapses trace diversity. The tolerance $\varepsilon_{\mathrm{MAE}}$ instead admits near-correct and physically plausible traces, which preserves multiple valid pathways, improves calibration, and lowers violation rates. The halting logic further reduces redundant sampling and concentrates supervision on diverse, high-fidelity trajectories.

4.3 Discussion

This result clarifies why general LRM training strategies are hard to directly transfer to process aware property prediction. Recent work [18] shows that curated rationales and test time scaling can yield strong results even when only about 53% of traces are correct, likely because tasks such as math, coding, and QA have binary correctness and explicit derivations. For property prediction, categorical correctness is a weak proxy for supervision quality because two traces may follow different discrete steps yet end within 1% of the true EQE, which is the signal the model must learn. OpenThoughts [60] finds that multi sampling often helps by preserving reasoning diversity, but for recipe to property the same unfiltered diversity amplifies label and trace noise, increases physics violation rate and weakens calibration. `PaRS` reconciles these views by keeping diversity where it matters, namely multiple near correct physical pathways, while trimming supervision to numerically consistent traces that respect simple physics. The result is a better accuracy and admissibility trade-off with about $15\times$ fewer traces than the multi sampling baseline.

5 Conclusion

We cast recipe to property prediction for materials discovery as a reasoning problem and introduce *Physics-aware Rejection Sampling (PaRS)* to curate supervision signals that are numerically accurate, calibrated, and physically admissible. `PaRS` replaces binary correctness and generic reward models with domain-grounded gates—range checks, a recipe-specific physical envelope, and a continuous error tolerance and adds variance and improvement-based halting. Instantiated with a QWEN3-235B teacher and a QWEN3-32B student on QD-LED device recipes, `PaRS` consistently optimize teacher-trace quality and yields student LRMs with lower MAE, higher correlation, better calibration, and markedly fewer physics violations. These gains are achieved with substantially less sampling, indicating a favorable quality–efficiency trade-off. While our experiments focus on EQE in QD-LEDs, the framework naturally extends to other device-level properties (e.g., lifetime, luminance) and to materials systems beyond optoelectronics. An important next step is to evaluate `PaRS` at larger scale and across diverse domains. We also envision coupling `PaRS` with RL-based adaptive exploration to enable closed-loop recipe design, where models not only predict reliably but also guide autonomous exploration of materials space under explicit physical guarantees.

References

[1] Eric A. Stach, Brian L. DeCost, A. Gilad Kusne, Jason Hattrick-Simpers, Keith A. Brown, Katia G. Reyes, Joshua Schrier, Simon J. L. Billinge, Tonio Buonassisi, and Ian Foster. The rise of self-driving laboratories in chemical and materials sciences. *Nature Synthesis*, 1:774–786, 2022. 1

[2] Manuel M. Flores-Leonar, Giovanni F. Caramori, Nancy A. Romero, et al. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(14):10137–10190, 2024. 1

[3] Peter I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. 1

[4] Elif Sabanzagil et al. Multi-fidelity bayesian data-driven design of energy absorbing spinodoid cellular structures. *arXiv preprint arXiv:2507.22079*, 2025. 1

[5] Ryan Jacobs, Logan Ward, Anubhav Jain, et al. Accurate predictions, uncertainty estimates, and accessibility: A roadmap for machine-learning-enabled materials discovery. *Machine Learning: Science and Technology*, 5(4):042501, 2024. 1

[6] Vladimir Varivoda, Kunpeng Dong, Shahrzad Omee, Zhenpeng Hu, et al. Materials property prediction with uncertainty quantification: A benchmark study. *arXiv preprint arXiv:2211.02235*, 2022. 1

[7] Tian Xie and Jeffrey C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical Review Letters*, 120:145301, 2018. 1

[8] Chi Chen, Weike Ye, Yifei Zuo, Chen Zheng, and Shyue Ping Ong. Graph networks as a universal machine learning framework for molecules and crystals. *Chemistry of Materials*, 31(9):3564–3572, 2019. 1

[9] Rubungo Andre Ndayishimiye et al. Llm4mat: Benchmarking large language models for materials science reasoning. *arXiv preprint arXiv:2507.14491*, 2025. 1, 2

[10] Zhe Liu, Nicholas Rolston, Austin C. Flick, Thomas W. Colburn, Zekun Ren, Reinhold H. Dauskardt, and Tonio Buonassisi. Machine learning with knowledge constraints for process optimization of open-air perovskite solar cell manufacturing. *Joule*, 6(4):834–849, 2022. 1

[11] Yao Lu, Dong Wei, Wu Liu, Juan Meng, Xiaomin Huo, Yu Zhang, Zhiqin Liang, Bo Qiao, Suling Zhao, Dandan Song, and Zheng Xu. Predicting the device performance of perovskite solar cells from the experimental parameters through machine learning of existing experimental results. *Journal of Energy Chemistry*, 77:200–208, 2023. 1

[12] Jiahao Xie, Yansong Zhou, Muhammad Faizan, Zewei Li, Tianshu Li, Yuhao Fu, Xinjiang Wang, and Lijun Zhang. Designing semiconductor materials and devices in the post-moore era by tackling computational challenges with data-driven strategies. *Nature Computational Science*, 4(5):322–333, 2024. 1

[13] Quanlong Guo et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 3

[14] Aaron Jaech, William Cohen, et al. The o1 system: Towards reasoning in large language models. *arXiv preprint arXiv:2412.16720*, 2024. 2, 3

[15] An Yang, Jiaqi Bai, and Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 2, 3

[16] National Research Council. *Integrated Computational Materials Engineering: A Transformational Discipline for Improved Competitiveness and National Security*. National Academies Press, Washington, DC, 2008. 2

[17] Wayne Xin Zhao, Kun Zhou, Junyi Li, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2024. 2

[18] Niklas Muennighoff et al. S1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025. 2, 3, 5, 8

[19] Eric Zelikman, Yuhuai Wu, Francis Sha, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In *NeurIPS*, 2022. 2, 5

[20] Zheng Yuan, Hongyi Yuan, Chengpeng Li, et al. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023. Introduces Rejection sampling Fine-Tuning (RFT). 2, 5

[21] Zhifeng Dong et al. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2309.17443*, 2023. 2, 5

[22] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. Introduces the GSM8K dataset. 2

[23] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, et al. Judging LLM-as-a-judge with LLMs. *arXiv preprint arXiv:2306.05685*, 2023. 2

[24] Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022. 2, 5, 6

[25] First name initial. Xu et al. Towards large reasoning models: A survey. *arXiv preprint*, 2025. Please replace with the exact arXiv ID when finalized. 2

[26] First name initial. Chen et al. Towards training large reasoning models. *arXiv preprint*, 2025. Please replace with the exact arXiv ID when finalized. 2

[27] X.-D. Xiang, X. D. Sun, et al. A combinatorial approach to materials discovery. *Science*, 268(5219):1738–1740, 1995. 2

[28] Ichiro Takeuchi, Robert Bruce van Dover, and Hideomi Koinuma. Combinatorial synthesis and evaluation of functional inorganic materials using thin-film techniques. *MRS Bulletin*, 27(4):301–308, 2002. 2

[29] Ben Ren et al. Inverse deep learning methods and benchmarks for artificial electromagnetic material design. *arXiv preprint arXiv:2112.10254*, 2021. 2

[30] Dianjing Liu, Yixuan Tan, Erfan Khoram, and Zongfu Yu. Training deep neural networks for the inverse design of nanophotonic structures. *ACS Photonics*, 5(4):1365–1369, 2018. 2

[31] Zhen Yang, Yoonho Kim, Yang Shao-Horn, et al. Inverse design with deep generative models: Next step in materials discovery. *National Science Review*, 9(8):nwac111, 2022. 2

[32] Santiago Miret and NM Anoop Krishnan. Are llms ready for real-world materials discovery? In *AI for Accelerated Materials Design-Vienna 2024*. 2

[33] Shuyi Jia, Chao Zhang, and Victor Fung. Llmatdesign: Autonomous materials discovery with large language models. *arXiv preprint arXiv:2406.13163*, 2024. 2

[34] Tanishq Gupta, Mohd Zaki, NM tAnoop Krishnan, and Mausam. Matscibert: A materials domain language model for text mining and information extraction. *npj Computational Materials*, 8(1):102, 2022. 2

[35] Yuwei Wan, Tong Xie, Nan Wu, Wenjie Zhang, Chunyu Kit, and Bram Hoex. From tokens to materials: Leveraging language models for scientific discovery. *arXiv preprint arXiv:2410.16165*, 2024. 2

[36] Junho Kim, Yeachan Kim, Jun-Hyung Park, Yerim Oh, Suho Kim, and SangKeun Lee. Melt: Materials-aware continued pre-training for language model adaptation to materials science. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10690–10703, 2024. 2

[37] Andre Niyongabo Rubungo, Craig Arnold, Barry P Rand, and Adji Bousso Dieng. Llm-prop: predicting the properties of crystalline materials using large language models. *npj Computational Materials*, 11(1):186, 2025. 3

[38] Youjia Li, Vishu Gupta, Muhammed Nur Talha Kilic, Kamal Choudhary, Daniel Wines, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Hybrid-llm-gnn: integrating large language models and graph neural networks for enhanced materials property prediction. *Digital Discovery*, 4(2):376–383, 2025. 3

[39] So Eun Choi, Miyoung Jang, SoHee Yoon, SangHyun Yoo, Jooyeon Ahn, Minho Kim, Ho-Gyeong Kim, Yebin Jung, Seongeon Park, Young-Seok Kim, et al. Llm-driven synthesis planning for quantum dot materials development. *Journal of Chemical Information and Modeling*, 65(6):2748–2758, 2025. 3

[40] Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance bayesian optimization. In *The Twelfth International Conference on Learning Representations*. 3

[41] Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with LLMs via bayesian optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. 3

[42] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 3

[43] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. 3

[44] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023. 3

[45] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023. 3

[46] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*. 3

[47] Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025. 3

[48] Eric Zelikman, Yuhuai Wu, and Noah D Goodman. Star: Self-taught reasoner. In *Proceedings of the NIPS*, volume 22, 2022. 3

[49] Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron C Courville, Alessandro Sordoni, and Rishabh Agarwal. V-star: Training verifiers for self-taught reasoners. *CoRR*, 2024. 3

[50] Woosung Koh, Wonbeen Oh, Jaein Jang, MinHyung Lee, Hyeongjin Kim, Ah Yeon Kim, Joonkee Kim, Junghyun Lee, Taehyeon Kim, and Se-Young Yun. Adastar: Adaptive data sampling for training self-taught reasoners. *arXiv preprint arXiv:2505.16322*, 2025. 3

[51] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2024. 3

[52] Yasuhiro Shirasaki, Geoffrey J. Supran, Moungi G. Bawendi, and Vladimir Bulović. Emergence of colloidal quantum-dot light-emitting technologies. *Nature Photonics*, 7:13–23, 2013. 3

[53] Bo Li, Fei Chen, Huaiyu Xu, Yang Song, Xiaohan Yan, Qiulei Xu, Longjia Wu, Yiran Yan, Wenjun Hou, Weiran Cao, Huaibin Shen, and Fengjia Fan. Advances in understanding quantum dot light-emitting diodes. *Nature Reviews Electrical Engineering*, 1:412–425, 2024. 3

[54] Sang Yun Bang, Yo-Han Suh, Xiang-Bing Fan, Dong-Wook Shin, Sanghyo Lee, Hyung Woo Choi, Tae Hoon Lee, Jiajie Yang, Shijie Zhan, William Harden-Chaters, Chatura Samarakoon, Luigi G. Occhipinti, Soo Deok Han, Sung-Min Jung, and Jong Min Kim. Technology progress on quantum dot light-emitting diodes for next-generation displays. *Nanoscale Horizons*, 6:68–77, 2021. 3

[55] Yu-Ming Huang, Konthoujam James Singh, An-Chen Liu, Chien-Chung Lin, Zhong Chen, Kai Wang, Yue Lin, Zhaojun Liu, Tingzhu Wu, and Hao-Chung Kuo. Advances in quantum-dot-based displays. *Nanomaterials*, 10(7):1327, 2020. 3

[56] Yasuhiro Shirasaki, Geoffrey J. Supran, Moungi G. Bawendi, and Vladimir Bulović. Emergence of colloidal quantum-dot light-emitting technologies. *Nature Photonics*, 7(1):13–23, 2013. 5

[57] Malte C. Gather and Sebastian Reineke. Recent advances in light outcoupling from white organic light-emitting diodes. *Journal of Photonics for Energy*, 5:057607, 2015. 5

[58] Light extraction analysis and enhancement in a quantum dot light emitting diode. *Optics Express*, 22(S7):A1783–A1798, 2014. rigorous dipole model for QLED outcoupling. 5

[59] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023. 6

[60] Tanmay Guha et al. Openthoughts: A scaling recipe for test-time compute in reasoning. *arXiv preprint arXiv:2506.04178*, 2025. 6, 8

```
<Prompt>: Device recipe
<Response>: Model's reasoning trace + final prediction.

# Role
You evaluate QD-LED EQE prediction responses (especially reasoning trace) quality
    with following rubric. Judge only against the provided device prompt.

# Scoring rubric (0~10)
1. Groundedness to Prompt (0~2.5): Quote prompt substrings for all used parameters;
    mark extra info as Assumption.
- 0.0~0.5: Largely ungrounded; few/no quotes; multiple unstated details.
- 0.6~1.3: Some quotes, but several parameters not cited; occasional unstated claims
- 1.4~2.0: Mostly grounded; 1-2 minor misses; assumptions called out but one is
    vague
- 2.1~2.3: Fully grounded with trivial omissions only
- 2.4~2.5: Every device parameter quoted; zero unstated details

2. Causal Reasoning Quality (0~2.0): Link given factors -> mechanisms -> EQE impact;
     separate Given / Inference / Implication.
- 0.0~0.4: Descriptive or hand-wavy; leaps from factors to EQE without mechanism.
- 0.5~1.0: Some correct factor->effect links but gaps and mixing of Given/Inference.
- 1.1~1.5: Coherent chains for most factors; clear separation with one notable gap
- 1.6~1.8: Mechanism-first, no unjustified jumps; discusses main loss channels
- 1.9~2.0: Exemplary: prioritizes the limiting mechanism.

3. Numerical & Unit Discipline (0~2.0): Show steps; keep %/nm/eV consistent;
    sensible rounding of final EQE.
- 0.0~0.4: Arithmetic or unit errors ; missing key steps.
- 0.5~1.0: Mostly correct; one error or unit slip.
- 1.1~1.5: Correct math; consistent units; minor omission .
- 1.6~1.8: Fully worked steps (e.g., IQE x outcoupling); sanity checks.
- 1.9~2.0: Clean, reproducible pipeline; precision noted where relevant.

4. Assumption Quality (0~2.0): Assumptions explicit, minimal, non-contradictory,
    each briefly justified.
- 0.0~0.4: Many hidden or contradictory assumptions.
- 0.5~1.0: Several assumptions; some lack justification.
- 1.1~1.5: Only necessary assumptions; short, credible justifications.
- 1.6~1.8: Minimal & well-justified; references common baselines.
- 1.9~2.0: Parsimonious and transparent; each assumption tied to its EQE impact;
    brief sensitivity note if applicable.

5. Clarity & Structure (0~1.5): Use sections: Given / Assumptions / Reasoning /
    Result; keep high signal-to-noise.
* 0.0~0.3: Disorganized; sections missing; EQE result absent or hard to find.
* 0.4~0.7: Sections present but uneven; some redundancy; result line imprecise.
* 0.8~1.1: Clear sections; stepwise logic; minor verbosity or formatting slips.
* 1.2~1.3: Crisp, concise, well-formatted; Result line prominent.
* 1.4~1.5: Polished, minimal, easy to audit; bullets/tables used judiciously.
```

Figure 5: `Prompt` for evaluating reasoning traces with DeepSeek-R1. We report the sum of average score of the five metrics to Table 1.

A Implementation details

A.1 Sampling schedule

For an example (x_i, y_i) and rounds $r = 1, 2, \ldots$ with mini-batch indices $j = 1, \ldots, b$, draw candidates $(\tau^{(r,j)}, \hat{y}^{(r,j)})$ and define the per-candidate error $e_{r,j} \coloneqq |\hat{y}^{(r,j)} - y_i|$. For each round, let $\bar{e}_r \coloneqq b^{-1} \sum_{j=1}^{b} e_{r,j}$, $s_r^2 \coloneqq (b-1)^{-1} \sum_{j=1}^{b} (e_{r,j} - \bar{e}_r)^2$, and $e_r^\star \coloneqq \min_{1 \le j \le b} e_{r,j}$; for $r \ge 2$,

define the improvement $\Delta_r := e^\star_{r-1} - e^\star_r$. The total candidate budget is $K_{\max}$, and T_r denotes the sampling temperature in round r.

A candidate is accepted if it satisfies all gates in Eqs. (2)–(4), namely $\hat{y}^{(r,j)} \in [0, 100]$, $|\hat{y}^{(r,j)} - y_i| \leq \varepsilon_{\mathrm{MAE}}$, and $\hat{y}^{(r,j)} \leq U(x_i)$. If multiple candidates pass within the same round, accept the one with the smallest j and terminate for that example.

If no candidate is accepted in round r, the procedure halts early under any of the following conditions: variance small enough, $s_r^2 \leq \varepsilon_{\mathrm{var}}$ (available from $r = 1$); lack of improvement, $\Delta_r \leq \delta_{\mathrm{imp}}$ (available from $r = 2$); or budget exhausted, $r\, b \geq K_{\max}$.

The temperature follows a capped, monotone increase to encourage exploration, for example $T_{r+1} = \min\{T_{\max}, \gamma T_r\}$ with $\gamma > 1$, or $T_{r+1} = \min\{T_{\max}, T_r + \Delta T\}$ with $\Delta T > 0$, starting from $T_1 = T_{\min}$.

The overall procedure is: at round $r = 1$, sample b candidates at temperature T_r and apply the acceptance gates; if none pass, compute s_r^2, $e_r^\star$, and (for $r \geq 2$) Δ_r; if no halting condition triggers, update the temperature according to the schedule and continue to $r+1$. Discard the example if no candidate is accepted before the budget is spent.

A.2 Prompt for LLM-as-a-Judge

We use DEEPSEEK-R1-0528-671B as the LLM-as-a-Judge. The prompt in Fig. 5 instructs the judge to evaluate synthesized reasoning traces against five rubrics and to return a numeric score for each rubric on a 0–10 scale with a brief justification. For the *LLM-as-a-Judge* selection baseline, we score all candidate traces generated for a prompt, compute a composite judge score by averaging the five rubric scores, and select the best one per prompt. For the summary metric reported in Table 1, we evaluate the set of traces selected by each method with the same judge prompt. We compute the composite score for each trace as the mean over the five rubrics and then report the mean across all evaluated prompts. This yields a single 0–10 score per method that is comparable across rejection sampling methods.

A.3 Token accounting for trace selection

For each prompt, let $T_{\text{teach,in}}$ and $T_{\text{teach,out}}$ denote the teacher's input tokens and *average* output tokens per generated trace. Let K be the generation budget, G the random number of traces actually generated before acceptance or budget exhaustion, and $K_{\text{avg}} := \mathbb{E}[G]$. Let T_{select} capture any extra token cost due to a selection pass (if present). Finally, let $r_{\text{acc}} \in [0, 1]$ be the probability that a prompt yields at least one accepted trace.

Expected tokens per prompt. The expected token cost per prompt, regardless of whether a trace is accepted, is

$$\mathbb{E}[\text{tokens per prompt}] = K_{\text{avg}}\big(T_{\text{teach,in}} + T_{\text{teach,out}}\big) + T_{\text{select}}. \tag{5}$$

For offline selection methods (random, self-consistency, and ours), the selection pass is negligible, so $T_{\text{select}} \approx 0$. By contrast, LLM-AS-A-JUDGE performs an additional inference pass over the concatenated set of generated traces. Approximating the judge pass as comparable in length to the teacher pass yields

$$\mathbb{E}[\text{tokens per prompt}]_{\text{judge}} \approx 2\, K_{\text{avg}}\big(T_{\text{teach,in}} + T_{\text{teach,out}}\big). \tag{6}$$

Expected tokens per accepted trace. When some prompts produce no accepted trace, it is useful to normalize by the acceptance rate r_{acc}. The expected tokens per accepted trace are

$$\mathbb{E}[\text{tokens per accepted}] = \frac{\mathbb{E}[\text{tokens per prompt}]}{\mathbb{E}[\text{accepted traces per prompt}]} \approx \frac{K_{\text{avg}}\big(T_{\text{teach,in}} + T_{\text{teach,out}}\big) + T_{\text{select}}}{r_{\text{acc}}}, \tag{7}$$

with the judge variant obtained by substituting Eq. (6) into Eq. (7).

Under online acceptance (our method), generation halts immediately upon acceptance or when the budget K is reached. Thus G follows a truncated geometric-like process, and K_{avg} reflects both early

acceptance on easy prompts and full-budget usage on hard prompts. Methods that commit to a fixed K without early stopping have $K_{\text{avg}} \approx K$.

In compute versus accuracy frontiers (e.g., Fig. 4), the x-axis reports the per-prompt token cost in Eq. (5). When comparing methods with materially different r_{acc}, we additionally report the per-accepted-trace cost using Eq. (7).

As a concrete example, suppose $T_{\text{teach,in}}$=900, $T_{\text{teach,out}}$=2000, K_{avg}=6.4, and r_{acc}=0.8. Then the per-prompt cost for offline selection is $6.4 \times (900 + 2000) = 18{,}560$ tokens. The judge variant is about $2 \times 18{,}560 = 37{,}120$ tokens per prompt. Normalizing by acceptance rate, the per-accepted-trace costs are approximately $23{,}200$ (offline) and $46{,}400$ (judge).