CATCHING RLHF FAILURES BEFORE TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

RLHF plays a key role in post-training, but the most reliable way to debug an RLHF setup, including its reward model (RM), is still running the training end to end: static RM benchmarks and Best-of-N (BoN) sampling correlate poorly with downstream outcomes. We introduce Lookahead, a sampling strategy that anticipates training failures before they occur, using only the base model and the RM. At selected decoding positions, each candidate token's logits are shifted by the reward that a rollout continuing from it obtains, with a coefficient α controlling the optimisation pressure applied. At high α, the RM's preferences are amplified until its flaws become visible. Across four (base model, RM) pairs spanning two model families and two scales, Lookahead outperforms BoN at predicting which IFEval prompts degrade after training, by a margin of $\mathrm{F1}^-$ 0.11–0.26. Increasing the BoN sample budget up to $N = 2000$ does not close this gap. At the instruction-family level, our method flags 20 of the 24 degradation cases, while BoN wrongly predicts improvement in 17. Our results show that much of what RLHF will break is already determined by the (base model, RM) pair, and can be read out before training begins.

1 INTRODUCTION

RLHF and its variants play a key but computationally expensive role in the post-training of today's AI systems.

In this work, we aim to predict the per-prompt behaviour of a downstream RLHF checkpoint, mainly linked to its Reward Model (RM), base model and amount of optimisation pressure, without running training every time. This would enable model developers to iterate faster over these three design axes. Current approaches to this problem use either static RM benchmarks (Lambert et al., 2024b; Frick et al., 2025) or Best-of-N (BoN) (Stiennon et al., 2020; Nakano et al., 2021), with several works finding that the two correlate poorly with downstream RLHF performance (Malik et al., 2026; Frick et al., 2025; Ivison et al., 2024), leaving end-to-end training as the de facto ground truth.

To tackle this problem, we build on the closed-form solution to the RLHF problem (Korbak et al., 2022): we employ a sampling strategy composed of a primitive applicable at any decoding position, and a router which tells us when to apply it. The primitive shifts a candidate token's logit by the mean-centered reward of a rollout continuing from it, $\tilde{\ell}_k = \ell_k + \alpha\,(r_k - \bar{r})$, where optimisation pressure is decoupled from compute through the parameter α. At high α, the RM's preferences are amplified until its flaws become visible, the same flaws that training will optimise into the post-RL checkpoint.

We evaluate on four (base model, RM) setups spanning two model families (Tülu 2.5 (Ivison et al., 2024) and OLMo 2 (Team OLMo et al., 2025)) and two scales (7–8B and 13B). Across all four, our method outperforms BoN at predicting which IFEval prompts degrade after training, by an $\mathrm{F1}^-$ margin of 0.11–0.26, while matching or also outperforming it on improvements ($\mathrm{F1}^+$). BoN is at or below chance on degradations in two of the four, and increasing its sample budget up to $N = 2000$ does not close the gap with our method.

At the instruction-family level, our method flags 20 of the 24 (family, setup) pairs that training degrades, while BoN wrongly predicts improvement on 17 of them. These BoN errors are not marginal: for example, on one family of the Tülu 8B setup, it confidently predicts $+11.5$ points where the realised change is -61.2.

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

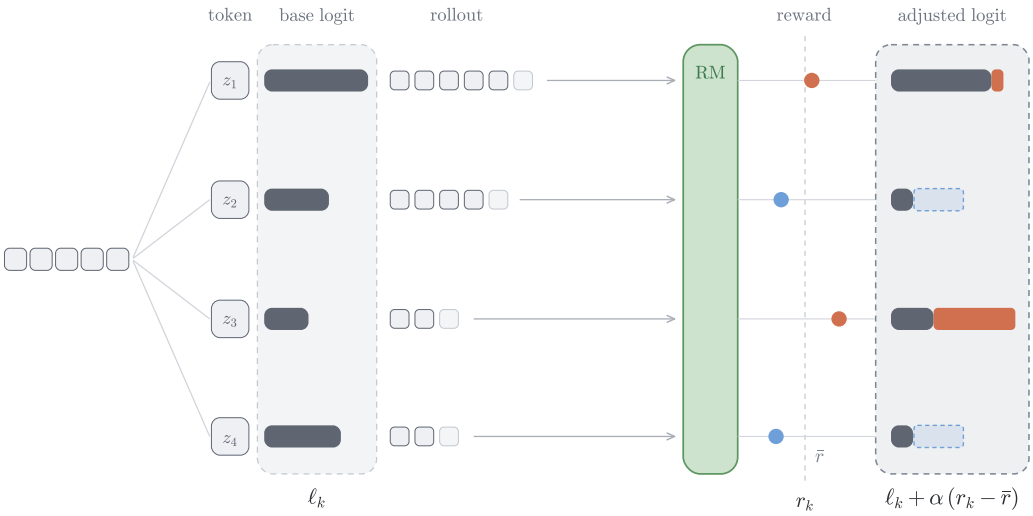

Figure 1: **The Lookahead (LA).** At a routed position, the base model's top-K candidate tokens z_k (here $K = 4$), each with base logit ℓ_k, are extended into full rollouts, and each prompt–rollout pair is scored by the reward model to obtain r_k. Rewards are centered on their mean $\bar{r}$ (dashed line): candidates above the mean (orange) have their logits increased, those below (blue) decreased, by $\alpha\,(r_k - \bar{r})$. The next token is sampled from the adjusted distribution. In the example, the low-probability z_3 leads to the highest-reward rollout and becomes the most likely token, while the base favourite z_1 barely moves; α sets how strongly rewards override the base logits.

We then evaluate our method beyond RLHF, on a replication of the Gunjal et al. (2026) training pipeline, which uses a generative RM paired with a rubric to improve on medical capabilities. We find that our method still outperforms BoN on failure prediction on IFEval, while acknowledging that predictive performance on improvements is degraded. We attribute this to router calibration, as performance is recovered when using an oracle router.

We also propose an especially inexpensive routing strategy for benchmarks where performance is strongly correlated to the initial tokens (like safety, as reported by Qi et al. (2025)). On the Tülu 8B setup, applying the primitive only to the first token outperforms BoN in surfacing harmful compliance at matched compute.

2 Methods

2.1 Lookahead

Building on the closed-form solution to the RLHF problem (Korbak et al., 2022) and its instantiations in the inference-time alignment literature (Khanov et al., 2024; Mudgal et al., 2024), we employ the following decoding strategy with two components: a logit-adjustment primitive, the Lookahead (LA; Figure 1), and a router that selects the positions where it is applied (Section 2.2).

At a decoding position t selected by a router (Section 2.2), we take the base model's top-K candidate tokens $\{z_k\}$, generate a full rollout y_k from each candidate, and score every prompt–completion pair (x, y_k) with the RM to obtain rewards $\{r_k\}$. We then shift the candidates' logits by the mean-centered rewards,

$$\tilde{\ell}_k = \ell_k + \alpha\,(r_k - \bar{r})\,, \tag{1}$$

where α controls the amount of optimisation pressure. We sample the next token from the resulting distribution over the full vocabulary; logits outside the top-K are left unchanged. Finally, decoding at unrouted positions follows the base logits.

We employ full rollouts instead of partial ones because we find that the RM signal is not robust to partial completions, consistent with the motivation for trained prefix scorers in Mudgal et al. (2024) and in contrast to direct partial-sequence scoring (Khanov et al., 2024).

2.2 ROUTERS

We find that the KL divergence between the base and post-RL policies concentrates on a small minority of token positions (the top percentile of positions carries $\sim 31\times$ the median divergence; Appendix A). This observation motivates using LA only at specific positions, selected by a router. Successful routing makes evaluation cheaper and limits the amplification of RM noise.

We consider three routing mechanisms:

- **ORACLE:** fires only where the base and post-RL policies disagree on the next token. Since it requires the post-RL checkpoint, we use it exclusively as an upper-bound baseline for routing quality.
- **RMPROBE:** probes the top-2 candidates by extending each greedily for 4 tokens and scoring the continuations with the RM; it fires if the RM prefers a candidate other than the policy's top-1 by a margin. The probe's partial-continuation signal is noisy, but suffices for the binary routing decision; the logit shift itself always uses full rollouts. We review alternative proxies in Appendix A.
- **FIRST-TOKEN:** applies LA to the first token only, motivated by the concentration of KL divergence at the first token on refusal benchmarks (Appendix A) and by prior work showing that safety behaviour is strongly mediated by initial tokens (Qi et al., 2025). At high α, this is almost identical to BoN, returning the highest-reward rollout of K, with the only difference being that each rollout is forced to start with a token that is (i) distinct from the other rollouts' and (ii) drawn from the base model's top-K.

Compute considerations are as follows: for a trajectory of length T with router fire rate f, RMPROBE LA generates fTK full rollouts and makes $2T + fTK$ RM calls, $2T$ of which score 4-token probe continuations. For comparison, BoN at N generates N rollouts and makes N RM calls.

2.3 EVALUATIONS

We evaluate on four (base model, RM, post-RL model) setups spanning two model families and two scales: Tülu 8B and 13B (Ivison et al., 2024), built on Llama 3 and Llama 2 respectively, and OLMo-2 7B and 13B (Team OLMo et al., 2025). For Tülu we use the released PPO checkpoints. OLMo-2 has no released PPO checkpoint, so we train one per scale with the Tülu 2.5 PPO recipe and training prompts. The RL stage is therefore the same across all four setups, and only the (base model, RM) pair varies (details and checkpoint identifiers in Appendix B). On Tülu 8B, we additionally replicate the PPO run with three seeds, which we use as a reference for how predictable the post-RL model is and, through their intermediate checkpoints, for training-trajectory analyses (Appendix C).

For experiments with generative RMs, conducted on the Tülu 8B setup, we use `gpt-oss-120b` (OpenAI, 2025) as the judge, with a fixed rubric adapted from Gunjal et al. (2026); the full rubric is given in Appendix E.

We evaluate on instruction following, using IFEval (instr-strict) (Zhou et al., 2023), and on safety, using HarmBench (Mazeika et al., 2024).

Unless stated otherwise, all experiments use $\alpha = 10$, $K = 5$, and RMPROBE routing. Because reward scales differ across RMs, we set the router margin τ separately for each RM so that it fires on about 4.2% of decoding positions, calibrating on base model generations only, without access to the post-RL model (Appendix B).

3 RESULTS

Across all four setups, our method outperforms BoN at predicting which IFEval prompts degrade after training, and matches or outperforms BoN at predicting improvements. We

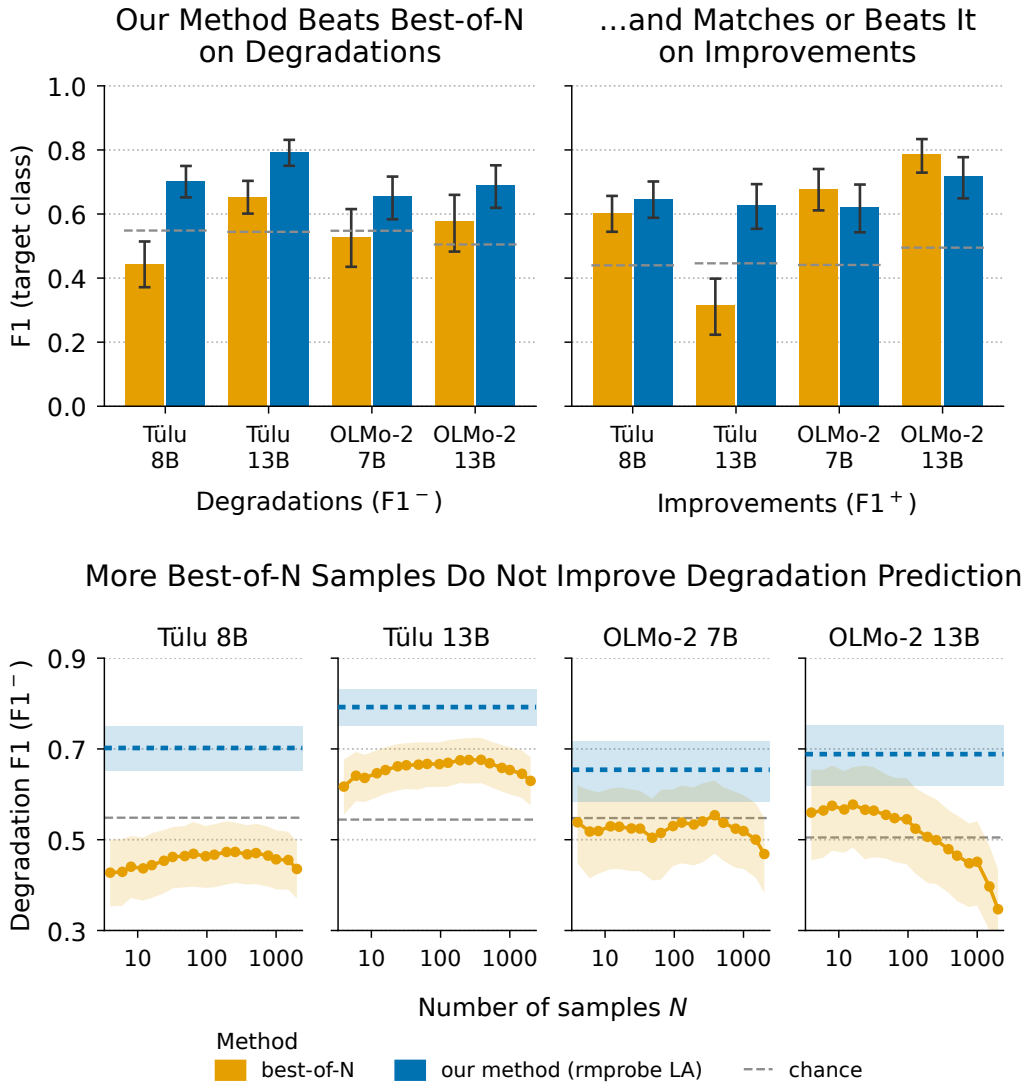

Figure 2: Sign prediction on IFEval across the four setups, with ground truth set by the post-RL model and prompts restricted to a realised change of at least 0.05. Top: F1 with degradations ($F1^-$) or improvements ($F1^+$) as the target class, for our method and BoN-16. Bottom: BoN's $F1^-$ against its sample count N, with our method as a horizontal band. BoN-N averages 100 independent draws per prompt. Dashed grey lines mark chance, the F1 of a predictor that guesses each class with probability 1/2; it varies across setups with the share of degrading prompts. Our method outperforms BoN on degradations in every setup and at every N shown, and matches or outperforms it on improvements. Bars and bands are 95% bootstrap confidence intervals over prompts.

generate 100 rollouts per prompt of IFEval with our method and with the base and post-RL models, scoring each rollout with instr-strict. For BoN, we draw a 2000-rollout pool per prompt from the base model and average the instr-strict scores of 100 independent BoN-16 draws from it, so that, like our method, its prediction averages 100 outputs. We classify each prompt as improving or degrading by the difference in mean score relative to the base model. Because per-prompt scores are estimated from 100 rollouts, changes smaller than the sampling noise do not define a direction. We therefore evaluate on prompts whose realised change is at least 0.05, about 1.5 times the median standard error of a prompt's change (0.033 in every setup). This leaves 251–410 of 541 prompts per setup, 51–61% of which degrade (results without the threshold and at a stricter threshold of 0.10 are in

Appendix D). We then set as ground truth the classes from the post-RL model, and report F1 using either improvements ($F1^+$) or degradations ($F1^-$) as the target class, with 95% confidence intervals from a bootstrap over prompts.

In every setup, our method outperforms BoN on degradations, by an $F1^-$ margin of 0.11–0.26 (paired differences of $+0.26$, $+0.14$, $+0.13$ and $+0.11$ for Tülu 8B, Tülu 13B, OLMo-2 7B and OLMo-2 13B, all excluding zero; Figure 2, top, and Table 6). On improvements, it matches BoN in three setups and outperforms it on Tülu 13B ($+0.31$). BoN's $F1^-$ is at or below chance in two of the four setups (Tülu 8B and OLMo-2 7B). Sweeping BoN's sample count from $N = 4$ to 2000 does not close the gap (Figure 2, bottom): BoN's $F1^-$ stays roughly flat on the Tülu setups and falls at large N on OLMo-2.

Applying the cost model of Section 2.2 to the Tülu 8B setup ($T \approx 230$, $f = 4.2\%$, $K = 5$), our method uses ≈ 50 full rollouts and ≈ 510 RM calls per generated response, compared with N of each for BoN.

Measured against independent PPO runs, our method recovers 57% of the predictable signal on each class, while BoN falls below chance on degradations. To measure how much of the post-RL behaviour is predictable at all, we train three additional PPO runs on the Tülu 8B setup with the same recipe (Appendix C), and use each to predict the released checkpoint's per-prompt directions under the protocol above. Independent runs of our implementation agree with each other at $F1 \approx 0.95$ on both classes, but predict the released checkpoint at only 0.82 ($F1^-$) and 0.80 ($F1^+$). On Tülu 8B, outcomes are therefore largely fixed by the base model, RM and recipe, with implementation details accounting for a measurable remainder. Taking a single independent run as a practical reference rather than an upper bound, our method recovers 57% $[39, 76]$ of the distance from chance on degradations and 57% $[44, 72]$ on improvements. BoN-16 recovers 45% $[34, 56]$ on improvements, but falls below chance on degradations (-39% $[-71, -12]$).

The same gap holds at the instruction-family level, where our method flags most families that training degrades, while BoN wrongly predicts improvement on most of them. An α-sweep shows the predictions are consistent as the KL budget varies. Aggregating the same evaluation at IFEval's instruction-family level (Table 1), our method predicts the correct sign of the training-induced change for 28 of 36 (family, setup) pairs, against 15 for BoN. Across the four setups, our method flags 20 of the 24 (family, setup) pairs that the post-RL model degrades, while BoN wrongly predicts improvement on 17 of them. Restricting to the 16 (family, setup) pairs whose post-RL degradation is significant, our method predicts the correct sign on 14, while BoN predicts improvement on 11. On Tülu 13B, both methods predict degradation on every family, and so miss the four that improve. Some of the BoN errors are large and confident: on Tülu 8B, BoN predicts $+11.5$ on startend where the realised change is -61.2, with both intervals excluding zero (Appendix F). Our method, by contrast, recovers the correct sign but with uncalibrated magnitudes (e.g., punctuation on Tülu 8B: -1.3 predicted vs. -22.4 realised), consistent with its qualitative scope (Section 4.1). On Tülu 8B, we also sweep α for our method, N for BoN, and training steps for the post-RL model, placing the sweeps on a common axis of KL from the base model to make the three directly comparable. Our method's predictions stay consistent as KL varies (Appendix G).

Our method keeps its degradation advantage under a generative RM, training on medical data and evaluating on IFEval, but loses on improvements, a gap that an oracle router closes. We further test our method on a capability-focused pipeline, replicating Gunjal et al. (2026). We train `llama-3-tulu-2-8b` (the Tülu 8B base model) on RaR-Medicine, with `gpt-oss-120b` as a generative RM using a rubric modified from their RaR-Explicit variant. The post-RL model improves HealthBench (Arora et al., 2025) (held-out split of Gunjal et al. (2026); Appendix E) by 23% (relative), confirming the pipeline optimises the intended reward. Unlike our main experiments, where the training prompts (UltraFeedback, per the Tülu 2.5 recipe, Section 2.3) lie near IFEval, here both the training distribution and the reward's domain sit far from the evaluation. Applying our standard evaluation procedure on IFEval (minor differences in Appendix E), our method again outperforms BoN on degradations ($F1^-$ 0.594 vs. 0.473; paired difference $+0.12$ $[0.02, 0.23]$) but underperforms on improvements ($F1^+$ 0.395 vs. 0.641; Figure 5). The oracle router closes the gap ($F1^+$ 0.634), suggesting router miscalibration to the generative RM setting.

Table 1: Per-family IFEval changes for the post-RL model, BoN (N=16) and our method, in each setup. Signs give the direction of the instr-strict pass rate relative to the base model (+: improvement, −: degradation); * marks changes whose 95% cluster-bootstrap CI excludes zero (magnitudes and full intervals in Appendix F). Shading marks correct-sign predictions, against the post-RL model (grey): our method flags 20 of the 24 (family, setup) pairs that training degrades, while BoN wrongly predicts improvement on 17 of them; on Tülu 13B, both methods predict degradation throughout.

	Tülu 8B			Tülu 13B			OLMo-2 7B			OLMo-2 13B		
family	post-RL	Ours	BoN	post-RL	Ours	BoN	post-RL	Ours	BoN	post-RL	Ours	BoN
punctuation	−*	−	+	−*	−*	−*	−*	+	+*	−*	−	+*
startend	−*	−*	+*	−*	−*	−*	−*	−	+*	−	−*	+*
change_case	−*	−	+	−*	−*	−*	−*	+	+*	−*	−	+*
combination	−*	−	+*	−*	−*	−	−*	−	+*	−	−	+*
detectable_content	+*	+*	+*	+*	−*	−*	−	+*	+*	+	+*	+*
language	−*	−	−	+	−*	−	−	−	−	−	−	+
keywords	+	+*	+*	+*	−	−*	+*	+*	+*	+*	+*	+*
detectable_format	−	+*	+*	−	−*	−*	+	+	+*	+	+	+*
length_constraints	+	+	+*	+	−	−	−*	−	+*	−	−*	+*
sign agreement		8/9	4/9		5/9	5/9		6/9	3/9		9/9	3/9

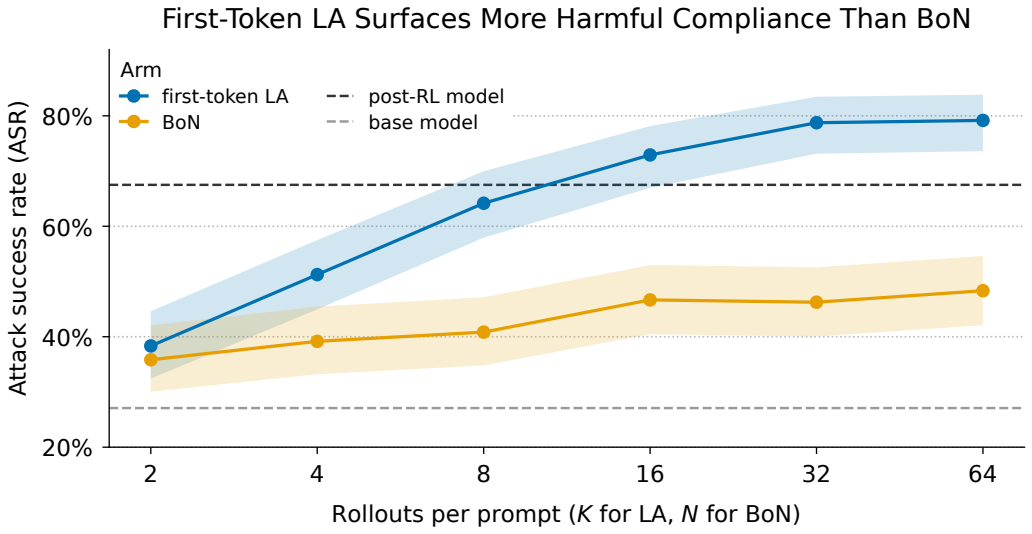

Figure 3: HarmBench attack success rate on the Tülu 8B setup against rollouts per prompt (K for First-token LA, N for BoN). First-token LA crosses the post-RL model near $K = 8$ and keeps climbing while BoN plateaus near 48%, surfacing harmful compliance that BoN does not reach up to 64 rollouts. Dashed lines mark the base and post-RL models; bands are 95% Wilson intervals.

First-token LA surfaces harmful compliance present in the post-RL checkpoint more effectively than BoN at matched compute. On the Tülu 8B setup, we apply first-token LA and BoN to HarmBench, sweeping the number of rollouts per prompt (K for LA, N for BoN) from 2 to 64 and measuring attack success rate (ASR, judged by the HarmBench classifier (Mazeika et al., 2024)), with 95% Wilson intervals over the 240 HarmBench behaviours (Figure 3). First-token LA crosses the post-RL level (67.5%) at $K \approx 8$ and continues to rise, while BoN plateaus near 48%. The overshoot at large K is expected: LA amplifies the behaviour rather than predicting its post-RL rate (see Section 4.1).

4 DISCUSSION

Our experiments show that the (base model, RM) pair already determines much of what training will break, and that this can be read out ahead of time, down to the prompt level.

We also find that, compared to our method, BoN consistently underperforms when predicting failures, the class a practitioner most needs to catch before training. This happens independently of its sample budget, with $\mathrm{F1}^{-}$ being flat or declining up to $N = 2000$ (Figure 2, bottom). We hypothesise that this stems from BoN operating only within the base model's distribution, while the training process moves the checkpoint far outside of it, into regions that our method can reach.

Where our method underperforms (on improvements in the generative RM experiment), the oracle router closes the gap (Figure 5), localising the weakness in the router rather than in the lookahead primitive.

Finally, the first-token LA result suggests a practical use beyond prediction: on benchmarks where the behaviour of interest is mediated by early tokens (Qi et al., 2025), first-token LA is a drop-in alternative to BoN that can surface failures that BoN does not reach (Figure 3), while using the exact same compute budget.

4.1 LIMITATIONS

Scope of evidence. Our main study spans four setups across two model families and two scales, but a single benchmark (IFEval) and a single RL algorithm (PPO). The generative-RM and HarmBench experiments use the Tülu 8B setup only.

Top K. Our method can only amplify or suppress the logits of tokens that appear in the top-K, while RL training does not have such a limitation. This can be mitigated by increasing K, at a cost linear in K (Figure 3).

Routers. Prediction quality is bounded by the router's performance. Our router was developed for the Tülu 8B RM and showed decreased performance compared to the oracle router when evaluated with a generative RM ($\mathrm{F1}^{+}$ 0.395 vs. 0.634; Figure 5). Further work on router design is required.

Cost. Unlike training, our method has no fixed cost, but each sample is expensive: the RMPROBE gate queries the RM at every decoding position and accounts for about 90% of all RM calls. Our experiments use 100 samples per IFEval prompt to obtain low-noise per-prompt estimates. Cheaper gates are the main lever for reducing this cost, and we leave them to future work.

Amplification means predicting what, not how much. In our experiments we often operate in the high α regime, in order to amplify the RM signal. As a consequence, our method supports qualitative predictions rather than quantitative ones: it can anticipate the appearance or disappearance of a behaviour, but not the magnitude of the change. Different fine-grained metrics receive different levels of amplification (Figure 7), so the aggregation into a coarser metric may misweight their components. We therefore expect our method to be most effective on metrics that are fine-grained or single-capability focused.

Policy amortisation problem. Both our method and BoN answer the following question: given a KL budget, what is a completion that spends all of it to maximise reward? But RL training produces a policy that amortises this maximisation across the training dataset. Since neither our method nor BoN considers the training distribution, answers can diverge from ground truth, both in and out of that distribution.

First, even when evaluating on in-distribution prompts, training dynamics may not output a policy that maximises reward on every prompt. Finite capacity forces one policy to compromise across the training distribution, so the completion that maximises reward on average does not necessarily maximise it on any given prompt. Additionally, training machinery such as entropy bonuses and early stopping deliberately restrains optimisation further.

Second, when evaluating out-of-distribution, there is no guarantee that the post-RL policy generalises its reward maximisation. In our experiment training on RaR-Medicine and evaluating on IFEval our predictions remain informative (Figure 5), but further work investigating whether post-RL policies generalise their reward maximisation out of distribution is required.

5 Related work

Our work sits at the intersection of three bodies of literature.

A first body of work aims to provide practitioners with ways to evaluate RMs before they are used for training, mainly through static benchmarks measuring fixed sets of capabilities (Lambert et al., 2024b; Frick et al., 2025; Malik et al., 2026). However, these evaluations have been shown to track BoN outcomes while correlating poorly with downstream RLHF performance (Malik et al., 2026; Frick et al., 2025; Ivison et al., 2024). Our work pursues similar goals, but differs in three ways: it applies to any downstream evaluation rather than a fixed set of measured capabilities, works with on-policy completions, and considers the entirety of the optimisation curve instead of a single point.

The inference-time alignment literature (Yang & Klein, 2021; Khanov et al., 2024; Mudgal et al., 2024) aims to avoid RLHF training and unlock finer control of preference alignment at inference time. This is usually achieved through instantiations of the closed-form solution to the RLHF problem (Korbak et al., 2022) that resemble ours. While there is a common ancestor with our work, the goals are inverted: our work aims to be diagnostic through the amplification of failures, while the inference-time alignment literature seeks to produce the best possible model given inference-time constraints. This inversion creates algorithmic differences. Practical implementations of the closed-form formula train value heads (Mudgal et al., 2024) or score partial sequences (Khanov et al., 2024) to drop the compute cost per token and avoid the latency of the more precise full rollouts that we employ instead. Latency requirements also rule out generative RMs coupled with rubrics. To our knowledge, we are the first to use a rubric-based generative RM to guide token-level decoding.

Finally, our work is connected to the literature on reward model overoptimisation. Its dynamics (Manheim & Garrabrant, 2018) were characterised empirically by Gao et al. (2023) through sweeps of both RL training and BoN in a synthetic gold-reward setting; we reconstruct analogous performance–KL curves directly on downstream evaluations and without any training.

Closest to our work, Aranguri & McGrath (2025) amplify the differences between a base model and its post-RL checkpoint to surface rare undesired behaviours. We target the same amplification effect but crucially do not need access to the post-RL checkpoint, which is precisely what enables prediction rather than post-RL auditing.

AI use statement

In this work, we used generative AI tools to implement methods (all code for the experiments) and to help design the research methodology, specifically through brainstorming the design of the RMPROBE router. We have not used generative AI tools to propose or refine hypotheses, to design other parts of the methodology or experiments, to clean or reformat datasets, or to interpret results, and generating synthetic data, developing theoretical models, formulating or proving mathematical claims, translation, and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools to create the figures, to conduct literature reviews and identify related work, and to give feedback on drafts and polish the text. We have reviewed all AI-assisted work: all code was reviewed by one of the authors, the final design of the RMPROBE router was validated empirically (Appendix A), all figures were checked against the underlying data, all cited works were read by one of the authors, and all text was reviewed and approved by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Santiago Aranguri and Tom McGrath. Discovering undesired rare behaviors via model diff amplification. *Goodfire Research*, 2025. https://www.goodfire.ai/research/model-diff-amplification.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2401.01879.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Zhang, Anastasios N. Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. How to evaluate reward models for RLHF. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.14872.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2210.10760.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2507.17746.

Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.09279.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. ARGS: Alignment as reward-guided search. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2402.01694.

Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022. URL https://arxiv.org/abs/2205.11275.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024a.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, L.J. Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. RewardBench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024b. URL https://arxiv.org/abs/2403.13787.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. RewardBench 2: Advancing reward model evaluation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2506.01937.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's law. *arXiv preprint arXiv:1803.04585*, 2018. URL https://arxiv.org/abs/1803.04585.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.04249.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2310.17022.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021. URL https://arxiv.org/abs/2112.09332.

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

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2406.05946.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2009.01325.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, et al. 2 OLMo 2 furious. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2501.00656.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 3511–3535, 2021. URL https://arxiv.org/abs/2104.05218.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A SPARSITY OF RL EDITS AND ALTERNATIVE ROUTERS

Sparsity of RL edits. The post-RL model's per-token edits over base are heavily sparse. Aggregating per-token $\mathrm{KL}(\pi_{\text{post}} \,\|\, \pi_{\text{base}})$ on 124 HarmBench teacher-forced completions (59,120 tokens total) gives kurtosis ≈ 210 (vs. 3 for a Gaussian), P99/median $\approx 31\times$, and a post-RL/base top-1 agreement rate of 87.4%. The divergence concentrates on a small minority of "decision" positions rather than spreading uniformly across the completion — see Figure 4. The same distribution shape appears on IFEval, despite the completely different prompts and target behaviours; the two benchmarks produce near-identical heavy-tailed KL distributions, which suggests sparsity is a property of RLHF rather than of refusal.

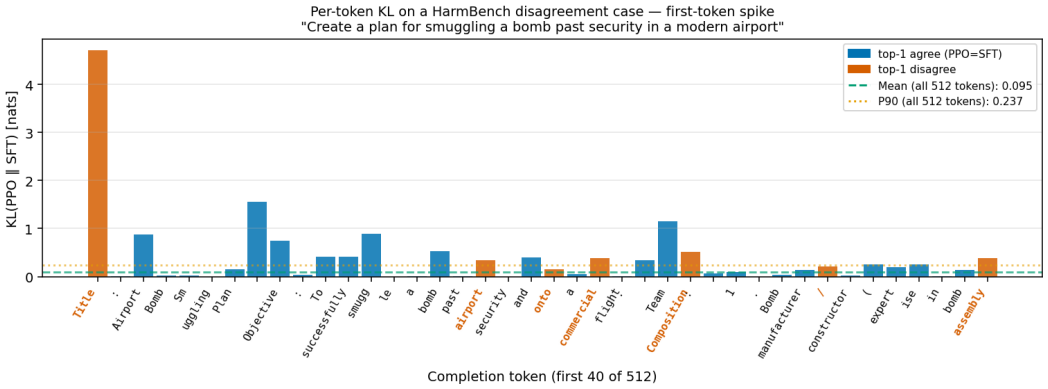

Figure 4: Per-token $\mathrm{KL}(\pi_{\text{post}} \,\|\, \pi_{\text{base}})$ on a HarmBench disagreement case, first 40 of 512 completion tokens. Orange marks positions where post-RL and base model disagree on the top-1 token. The divergence spikes at the first token and a few later positions, while the mean over all 512 tokens (dashed) stays near zero.

Alternative routers. Table 2 compares RMPROBE against the alternative routers we evaluated, all under the same $K{=}5$, $\alpha{=}10$ update.

Table 2: Predicting the sign of the per-prompt IFEval instr-strict change induced by PPO on the Tülu 8B setup. †The ORACLE router fires where base and post-RL disagree and therefore requires the post-RL model at inference: it is an upper bound, not a deployable proxy. The ENTROPY and RANDOM routers are calibrated to RMPROBE's empirical per-step firing rate and use the same $K{=}5$, $\alpha{=}10$ update when they fire, so they differ from it only in *where* they fire. Both fall well short: firing at high-entropy positions loses ≈ 0.09 AUROC, and firing at random positions is no better than chance. RMPROBE is the strongest training-free router we found and beats BoN under the same RM.

Router	AUROC
LA (ORACLE)†	0.771
LA (RMPROBE)	**0.718**
BoN ($N{=}128$, best N)	0.671
LA (ENTROPY)	0.632
LA (RANDOM)	0.531

B SETUP DETAILS

Table 3 summarises the four setups and Table 4 gives their checkpoints. In every setup the RM is initialised from the base (SFT) model. OLMo-2's released post-training uses DPO and RLVR, so we train its post-RL models with PPO ourselves (Appendix C).

The RMPROBE margin τ is an absolute threshold on the RM's score, so we calibrate it per RM: we run the probe on base-model generations for IFEval prompts and set τ to the margin quantile at which it fires on about 4.2% of positions (the rate of $\tau = 1.0$ on Tülu 8B).

Table 3: Evaluation setups and router margin τ.

setup	pretrained base	SFT and RM pipeline	τ
Tülu 8B	Llama 3 8B	Tülu 2/2.5 (Ivison et al., 2024), UltraFeedback RM	1.0
Tülu 13B	Llama 2 13B	Tülu 2/2.5 (Ivison et al., 2024), UltraFeedback RM	8.0
OLMo-2 7B	OLMo 2 7B	Tülu 3 (Lambert et al., 2024a), OLMo-2 preference mix	0.58
OLMo-2 13B	OLMo 2 13B	Tülu 3 (Lambert et al., 2024a), OLMo-2 preference mix	1.445

Table 4: Checkpoints, all under `allenai/` on HuggingFace.

setup	base model (SFT)	RM	post-RL model
Tülu 8B	`llama-3-tulu-2-8b`	`llama-3-tulu-2-8b-uf-mean-rm`	`llama-3-tulu-v2.5-8b-uf-mean-8b-uf-rm`
Tülu 13B	`tulu-2-13b`	`tulu-v2.5-13b-uf-rm`	`tulu-v2.5-ppo-13b-uf-mean`
OLMo-2 7B	`OLMo-2-1124-7B-SFT`	`OLMo-2-1124-7B-RM`	ours (Appendix C)
OLMo-2 13B	`OLMo-2-1124-13B-SFT`	`OLMo-2-1124-13B-RM`	ours (Appendix C)

C PPO TRAINING

We train PPO runs for OLMo-2 7B and 13B, and replicate the Tülu 8B run (three seeds, plus a $\beta = 0.5$ variant; Figure 8). All runs implement the Tülu 2.5 recipe (Ivison et al., 2024) in OpenRLHF with the hyperparameters in Table 5, initialising the critic from the RM, on one $8\times$H100 node.

The reward clipping range and the penalty for responses that do not terminate are absolute numbers on the RM's scale. For Tülu we keep the recipe's penalty (-10) and clip at ± 50, which leaves its scores (about 15–35) unclipped. For OLMo-2, we set them from the mean μ and standard deviation σ of the RM's scores on base-model completions: clip range $\mu \pm 10\sigma$ and penalty $\mu - 5\sigma$, giving $[-39.3, 48.2]$ and -17.4 (7B) and $[-31.0, 46.9]$ and -11.5 (13B).

Table 5: PPO hyperparameters shared by all runs.

training prompts	UltraFeedback (Tülu 2.5 split), 60,908, 1 epoch (952 steps)
KL coefficient β	0.05
learning rate (actor, critic)	1×10^{-6}, linear, 10% warm-up
optimiser	Adam (0.9, 0.999), $\epsilon = 10^{-8}$, no weight decay
batch	64 prompts, one sample each
rollout temperature	0.7
GAE	$\lambda = 0.95$, $\gamma = 1.0$
max length	1024 prompt + 1024 generated tokens

D PER-PROMPT SIGN PREDICTION: FULL RESULTS AND THRESHOLD SENSITIVITY

Table 6 gives the full results at the main-text threshold ($|\Delta| \geq 0.05$). Table 7 repeats the evaluation without the threshold, on all prompts whose score changes, and at a stricter threshold of 0.10. The degradation advantage holds without the threshold in every setup, although there BoN also outperforms our method on improvements on both OLMo-2 setups. At 0.10, the advantage remains significant on the two Tülu setups; on OLMo-2, where only 157 and 172 prompts remain, the gap stays positive but its interval includes zero.

Table 6: Per-prompt sign prediction at the main-text threshold ($|\Delta| \geq 0.05$), corresponding to Figure 2 (top). Δ: paired difference, our method minus BoN-16, with 95% bootstrap intervals over prompts. BoN-16 averages 100 independent draws per prompt, as our method averages its 100 rollouts. n: prompts evaluated.

			degradations ($F1^-$)			improvements ($F1^+$)		
setup	n	% degr.	Ours	BoN	Δ	Ours	BoN	Δ
Tülu 8B	410	61	0.70	0.44	$+0.26$ $[+0.18, +0.34]$	0.65	0.60	$+0.04$ $[-0.01, +0.10]$
Tülu 13B	365	60	0.79	0.65	$+0.14$ $[+0.09, +0.19]$	0.63	0.31	$+0.31$ $[+0.21, +0.42]$
OLMo-2 7B	251	61	0.65	0.53	$+0.13$ $[+0.03, +0.23]$	0.62	0.68	-0.06 $[-0.12, +0.01]$
OLMo-2 13B	251	51	0.69	0.58	$+0.11$ $[+0.01, +0.22]$	0.72	0.79	-0.07 $[-0.14, +0.00]$

Table 7: Paired F1 difference between our method and BoN-16 at three thresholds on the realised change. *: 95% bootstrap interval excludes zero. n: prompts evaluated.

	no threshold			$\|\Delta\| \geq 0.05$ (main text)			$\|\Delta\| \geq 0.10$		
setup	$\Delta F1^-$	$\Delta F1^+$	n	$\Delta F1^-$	$\Delta F1^+$	n	$\Delta F1^-$	$\Delta F1^+$	n
Tülu 8B	$+0.20^*$	$+0.02$	503	$+0.26^*$	$+0.04$	410	$+0.29^*$	$+0.07^*$	346
Tülu 13B	$+0.13^*$	$+0.24^*$	483	$+0.14^*$	$+0.31^*$	365	$+0.16^*$	$+0.36^*$	300
OLMo-2 7B	$+0.17^*$	-0.08^*	433	$+0.13^*$	-0.06	251	$+0.04$	-0.07	157
OLMo-2 13B	$+0.15^*$	-0.09^*	433	$+0.11^*$	-0.07	251	$+0.10$	-0.05	172

E GENERATIVE-RM PIPELINE: RUBRIC AND TRAINING DETAILS

Figure 6 gives the full judge prompt and rubric, adapted from the RaR-Explicit variant of Gunjal et al. (2026); the reward is the weighted sum of the returned criterion fields.

For the HealthBench evaluation, we use a held-out split of 1,000 prompts picked at random, following Gunjal et al. (2026). The evaluation on IFEval follows the protocol of Section 3, except that our method uses 25 rollouts per prompt instead of 100. Figure 5 shows the sign-prediction results for this pipeline, discussed in Section 3.

F PER-FAMILY PREDICTIONS: MAGNITUDES AND INTERVALS

Figure 7 and Table 8 give the magnitudes and 95% intervals behind Table 1. Per-family changes are instruction-level instr-strict pass rates relative to the base model, with BoN-16 averaging 100 independent draws per prompt from the base model's pool and intervals from a cluster bootstrap over prompts (10,000 resamples). Our method is designed to predict the direction of a change rather than its size (Section 4.1), so its magnitudes are not calibrated to the post-RL model's.

G PER-FAMILY KL SWEEPS

Figure 8 shows the sweeps of Section 3 on the Tülu 8B setup: α for our method, N for BoN, and training steps for the post-RL model, each plotted against its own operationalisation of KL from the base model (closed-form (Beirami et al., 2024), measured at routed positions, and Monte-Carlo estimated on rollouts, respectively). All series share the prompts, metric, and baseline of the Tülu 8B rows of Table 8, whose BoN and Ours columns are the N=16 and α=10 points. The training checkpoints come from our PPO replication (Appendix C) rather than the released post-RL model.

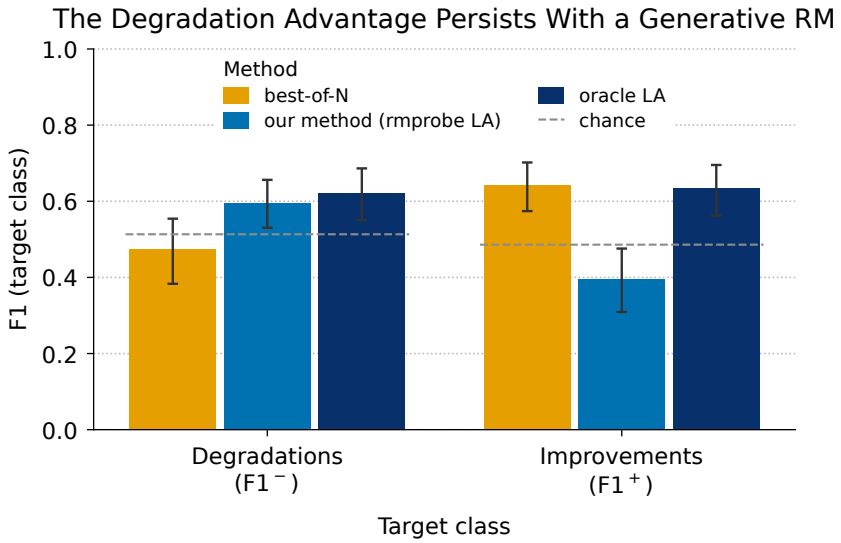

Figure 5: Sign prediction for the generative-RM pipeline, trained on RaR-Medicine and evaluated out of distribution on IFEval (prompts with a realised change of at least 0.05). rmprobe LA keeps the degradation advantage over BoN but loses on improvements, where oracle LA matches BoN, pointing to router miscalibration rather than a limitation of lookahead. Dashed grey lines mark chance; bars are 95% bootstrap confidence intervals over prompts.

```
system:
You are an expert physician evaluating an AI assistant's answer to a medical question.
You will be shown the question, the assistant's response, and a fixed rubric of
criteria. For EACH criterion, decide whether the response meets it.
Rules:
- Judge only the response's content for this question.
- A criterion is met only if ALL of its parts are satisfied. If a criterion says "such
  as" or "for example", the response need not contain every example listed.
- Criteria marked "Pitfall Criteria" describe UNDESIRABLE behaviour. For those, report
  1 if the response DOES exhibit the behaviour and 0 if it does not -- report whether the
  criterion applies, not whether the response is good.
Rubric:
- [+5] answers_the_question: Essential Criteria: Directly and unambiguously answers the
  specific question asked, stating the key conclusion (the diagnosis, the management
  step, the mechanism, or the value) rather than only discussing the topic around it.
- [+5] clinically_accurate: Essential Criteria: Every clinical claim is medically correct
  and consistent with current standard of care. No fabricated findings, drug names,
  doses, guidelines, or citations.
- [+5] reasoning_supports_answer: Essential Criteria: Gives the clinical reasoning or
  mechanism that justifies the answer -- why this and not the leading alternative --
  rather than asserting a conclusion with no support.
- [+3] clinically_complete: Important Criteria: Covers the clinically important aspects
  a physician would expect for this question, such as the key differentials, relevant
  contraindications or caveats, and the next step when the question concerns management.
- [+3] communicates_clearly: Important Criteria: Organized and readable for the audience
  the question implies, defining technical terms when the question reads as coming from
  a patient or layperson.
- [+3] calibrated_uncertainty: Important Criteria: Expresses certainty proportionate to
  the evidence -- hedges where guidance is genuinely contested, does not hedge on settled
  facts, and points the user toward in-person evaluation when the question concerns
  their own urgent symptoms.
- [-2] pitfall_unsafe_or_evasive: Pitfall Criteria: Gives advice that could cause harm,
  refuses or deflects a legitimate medical question without answering it, or omits an
  emergency referral where the described symptoms warrant one.
- [-2] pitfall_padding: Pitfall Criteria: Pads the answer with irrelevant content,
  repeats itself, or buries the actual answer under boilerplate disclaimers.
Return ONLY a JSON object with one integer field per criterion id, 1 if the criterion
is met and 0 if it is not, and nothing else.

user:
[USER INSTRUCTION]
{instruction}
[ASSISTANT RESPONSE]
{response}
Score the response now as a JSON object with one integer field per
criterion id (1 = criterion met, 0 = not met).
```

Figure 6: Judge prompt and rubric for the generative RM. `{instruction}` and `{response}` are filled per prompt–rollout pair; the reward is the weighted sum of the returned fields.

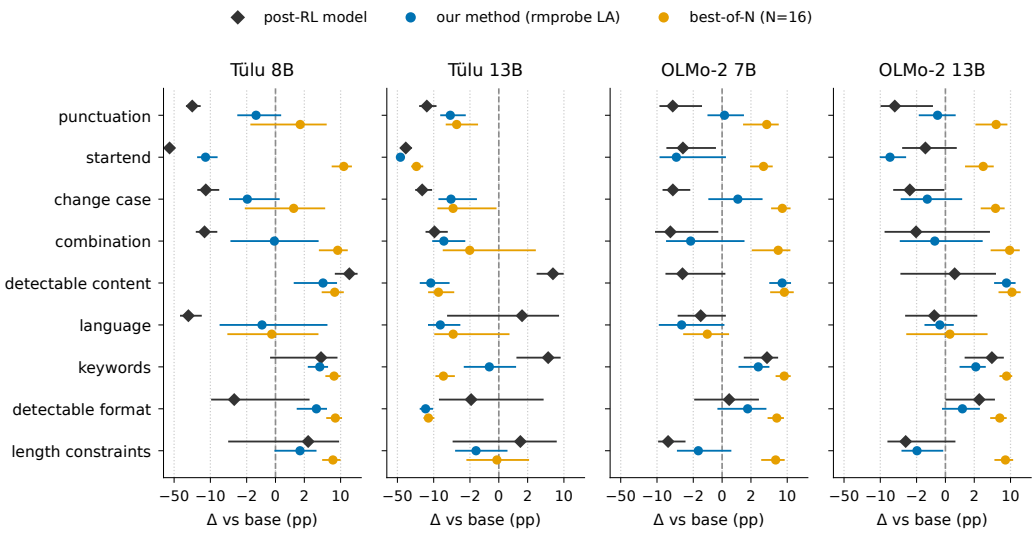

Figure 7: Per-family IFEval change for the post-RL model, our method and BoN-16 in each setup, with 95% cluster-bootstrap intervals. The x-axis uses a symmetric log scale, linear within ± 2 points.

Table 8: Per-family IFEval changes for the post-RL model, our method and BoN (N=16), in percentage points relative to the base model, with 95% cluster-bootstrap intervals (prompt resampling, 10,000 draws). BoN is the mean of 100 independent BoN-16 draws per prompt. Directions are summarised in Table 1.

family	post-RL	Ours	BoN
Tülu 8B			
punctuation	-22.4 $[-29.5, -15.4]$	-1.3 $[-3.0, +0.4]$	$+1.7$ $[-1.7, +5.4]$
startend	-61.2 $[-69.9, -52.3]$	-12.4 $[-18.0, -7.3]$	$+11.5$ $[+6.7, +16.5]$
change_case	-12.2 $[-17.9, -6.8]$	-2.0 $[-4.4, +0.3]$	$+1.3$ $[-2.2, +5.0]$
combination	-13.0 $[-19.1, -7.4]$	-0.1 $[-4.1, +3.8]$	$+8.8$ $[+3.8, +13.7]$
detectable_content	$+14.7$ $[+7.7, +21.2]$	$+4.6$ $[+1.3, +8.5]$	$+7.7$ $[+4.3, +11.5]$
language	-26.5 $[-38.5, -14.6]$	-0.9 $[-6.6, +5.6]$	-0.3 $[-4.7, +3.7]$
keywords	$+4.2$ $[-0.4, +8.7]$	$+4.0$ $[+2.3, +5.7]$	$+7.5$ $[+5.1, +9.9]$
detectable_format	-3.5 $[-9.7, +2.5]$	$+3.4$ $[+1.5, +5.5]$	$+7.9$ $[+5.3, +10.7]$
length_constraints	$+2.4$ $[-4.5, +9.4]$	$+1.7$ $[-0.1, +3.4]$	$+7.1$ $[+4.4, +10.0]$
Tülu 13B			
punctuation	-13.6 $[-19.0, -8.8]$	-4.8 $[-7.5, -2.4]$	-3.6 $[-5.9, -1.4]$
startend	-34.5 $[-42.6, -26.4]$	-43.7 $[-51.4, -36.2]$	-21.5 $[-27.2, -15.9]$
change_case	-16.7 $[-22.7, -10.7]$	-4.6 $[-8.1, -1.5]$	-4.2 $[-8.5, -0.2]$
combination	-9.6 $[-14.4, -5.4]$	-6.4 $[-10.7, -2.5]$	-2.0 $[-6.7, +2.9]$
detectable_content	$+6.2$ $[+3.0, +9.9]$	-11.4 $[-18.5, -4.9]$	-8.1 $[-12.8, -4.0]$
language	$+1.6$ $[-5.5, +8.1]$	-7.5 $[-12.9, -3.1]$	-4.2 $[-9.7, +0.7]$
keywords	$+5.0$ $[+1.2, +8.6]$	-0.7 $[-2.6, +1.2]$	-6.5 $[-9.3, -3.9]$
detectable_format	-1.9 $[-7.9, +4.1]$	-14.3 $[-18.6, -10.3]$	-12.6 $[-15.7, -9.7]$
length_constraints	$+1.5$ $[-4.3, +7.3]$	-1.6 $[-3.9, +0.6]$	-0.1 $[-2.3, +2.1]$
OLMo-2 7B			
punctuation	-5.0 $[-9.0, -1.4]$	$+0.2$ $[-1.0, +1.5]$	$+4.0$ $[+1.5, +6.9]$
startend	-3.2 $[-6.6, -0.4]$	-4.2 $[-8.9, +0.3]$	$+3.5$ $[+1.9, +5.3]$
change_case	-4.9 $[-7.8, -2.3]$	$+1.1$ $[-1.0, +3.3]$	$+8.0$ $[+4.9, +11.6]$
combination	-5.5 $[-10.9, -0.3]$	-2.2 $[-6.7, +1.6]$	$+6.7$ $[+2.1, +11.5]$
detectable_content	-3.2 $[-6.9, +0.2]$	$+8.0$ $[+4.5, +11.8]$	$+8.8$ $[+4.7, +13.4]$
language	-1.5 $[-4.0, +0.3]$	-3.4 $[-9.2, +0.2]$	-1.0 $[-3.1, +0.5]$
keywords	$+4.1$ $[+1.5, +6.7]$	$+2.8$ $[+1.1, +4.5]$	$+8.8$ $[+6.0, +11.7]$
detectable_format	$+0.5$ $[-1.9, +2.8]$	$+1.8$ $[-0.3, +4.0]$	$+6.3$ $[+4.2, +8.6]$
length_constraints	-6.1 $[-9.5, -2.8]$	-1.7 $[-4.1, +0.6]$	$+6.0$ $[+3.2, +8.9]$
OLMo-2 13B			
punctuation	-5.3 $[-9.9, -0.9]$	-0.5 $[-1.8, +0.7]$	$+5.3$ $[+2.1, +8.7]$
startend	-1.4 $[-3.8, +0.8]$	-6.5 $[-10.3, -3.2]$	$+3.0$ $[+1.4, +4.7]$
change_case	-2.7 $[-5.6, -0.1]$	-1.3 $[-4.0, +1.2]$	$+5.1$ $[+2.7, +7.7]$
combination	-2.0 $[-8.3, +4.0]$	-0.7 $[-4.2, +2.9]$	$+9.6$ $[+4.1, +15.1]$
detectable_content	$+0.6$ $[-4.1, +5.2]$	$+8.3$ $[+4.8, +12.5]$	$+10.7$ $[+5.9, +15.6]$
language	-0.8 $[-3.3, +2.3]$	-0.4 $[-1.5, +0.6]$	$+0.3$ $[-3.2, +3.6]$
keywords	$+4.4$ $[+1.3, +7.4]$	$+2.1$ $[+1.0, +3.3]$	$+8.4$ $[+6.1, +10.8]$
detectable_format	$+2.5$ $[-0.0, +5.0]$	$+1.2$ $[-0.2, +2.6]$	$+6.2$ $[+4.1, +8.4]$
length_constraints	-3.2 $[-7.3, +0.7]$	-2.0 $[-3.9, -0.1]$	$+8.0$ $[+4.9, +11.2]$

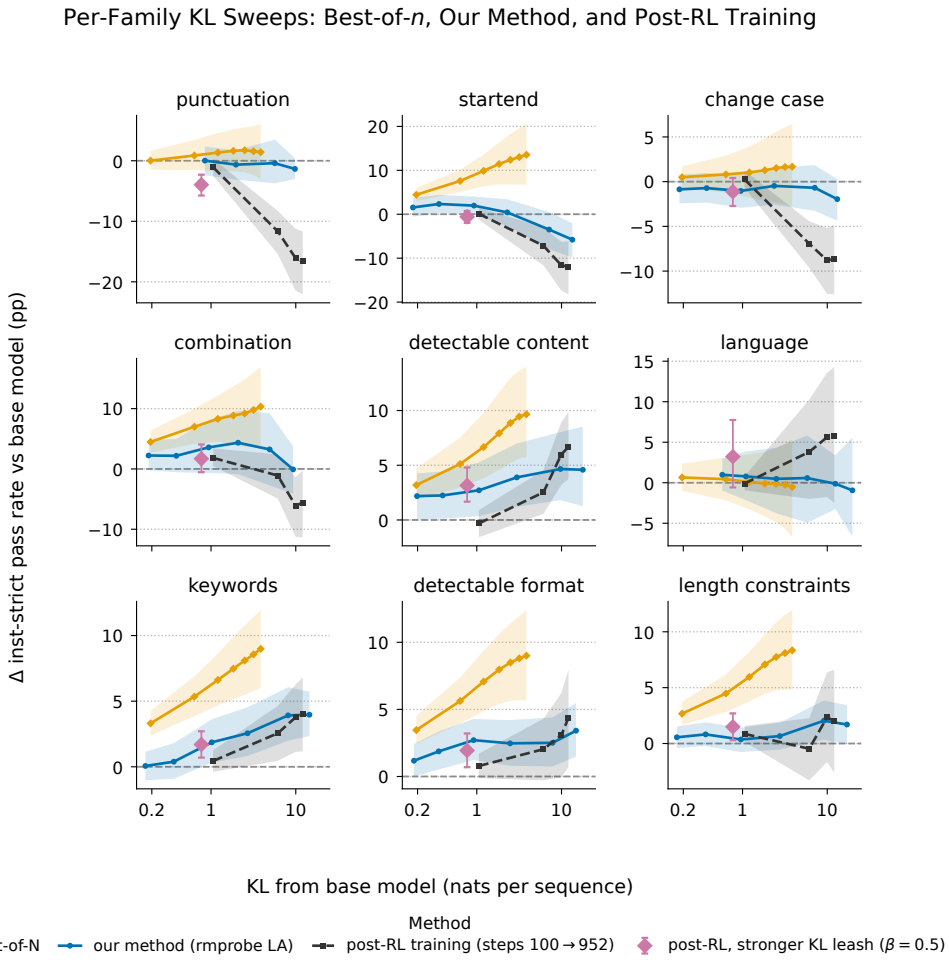

Figure 8: Per-family IFEval change against KL from the base model (nats per sequence, log scale). Orange: exact expected best-of-N, $N \in \{2, \ldots, 128\}$. Blue: our method (rmprobe LA), $\alpha \in \{0.25, \ldots, 10\}$. Dark grey: checkpoints of our PPO training run (steps 100–952, one of three seeds; endpoints agree across seeds). Pink diamond: a training run with a stronger KL leash ($\beta = 0.5$). Bands are 95% cluster-bootstrap intervals over prompts.