

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 FROM CURIOSITY TO CAUTION: MITIGATING REWARD HACKING FOR BEST-OF- N WITH PESSIMISM

Anonymous authors

Paper under double-blind review

ABSTRACT

Inference-time compute scaling has emerged as a powerful paradigm for improving language model performance on a wide range of tasks, but the question of how best to use the additional compute remains open. A popular approach is *Best-of- N* (BoN) sampling, where N candidate responses are generated, scored according to a reward model, and the highest-scoring response is selected. While this approach can improve performance, it is vulnerable to *reward hacking*, where performance degrades as N increases due to the selection of responses that exploit imperfections in the reward model instead of genuinely improving generation quality. Prior attempts to mitigate reward hacking—via stronger reward models or heavy-handed distributional regularization—either fail to fully address over-optimization or are too conservative to exploit additional compute. In this work, we explore the principle of *peSSImism* in reinforcement learning (RL), which uses lower confidence bounds on value estimates to avoid out-of-distribution (OOD) actions with uncertain reward estimates. Our approach, termed as *caution*, can be seen as the reverse of *curiosity*: where curiosity (e.g., via Random Network Distillation, RND) rewards prediction error as a signal of novelty, caution penalizes prediction error as a signal of distributional uncertainty. Practically, caution trains an error model on typical responses and uses its prediction error to lower reward estimates for atypical ones. Our extensive empirical evaluation demonstrates that caution is a simple, computationally efficient approach that substantially mitigates reward hacking in BoN sampling. We also provide a theoretical analysis in a simplified linear setting, which shows that caution provably improves over the standard BoN approach. Together, our results not only establish caution as a practical solution to reward hacking, but also provide evidence that curiosity-based approaches can be a general OOD detection technique in LLM settings.

1 INTRODUCTION

Inference-time scaling has emerged as a transformative paradigm for enhancing language model performance, enabling significant improvements across a wide range of reasoning tasks without increasing model size (Brown et al., 2024; Guo et al., 2025; Jaech et al., 2024). This success motivates the question of how best to leverage additional inference-time compute to maximize performance. A particularly popular and effective approach is *Best-of- N* (BoN) sampling (Stiennon et al., 2020; Nakano et al., 2021; Wang et al., 2022; Li et al., 2022; Huang et al., 2025a), where multiple candidate responses are generated for a given prompt, scored according to a reward model \hat{r} , and the highest-scoring response is selected. This approach capitalizes on the intuition that generating more candidates should increase the probability of finding higher-quality solutions, allowing the model to effectively ‘explore’ a larger portion of the response space than it could with only a single response.

While BoN is a simple and competitive baseline that is capable of astonishing gains in many settings (Brown et al., 2024), its success is fundamentally limited by the quality of the reward model \hat{r} used to score and select responses. Indeed, a common phenomenon often occurs with BoN, where performance initially improves as N increases, but then hits an inflection point after which larger N lead to increasingly worse outcomes (Gao et al., 2023; Huang et al., 2025b; Khalaf et al., 2025); an example of this phenomenon can be seen in Figure 1, where we plot the performance of BoN on the mathematical reasoning task GSM8K for different N scored by several different reward models. This counterintuitive phenomenon, whereby increasing N leads to worse performance, occurs due

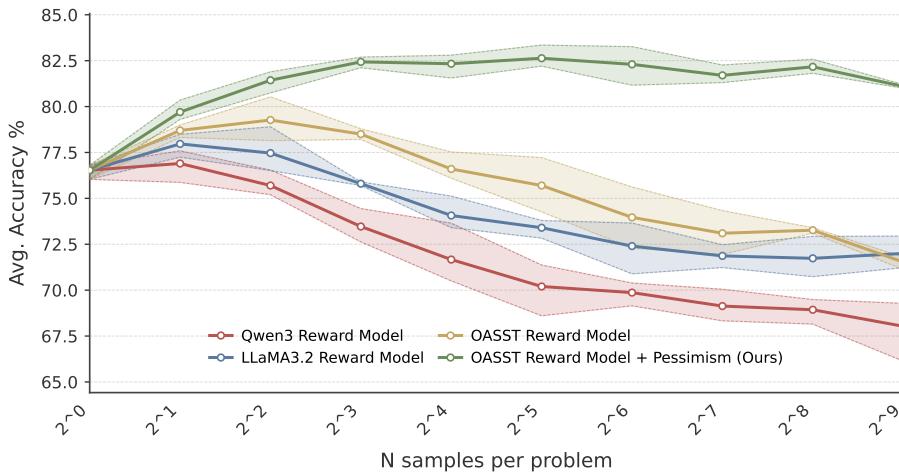


Figure 1: **Average Accuracy with different sampling budgets for Best-of- N** on the GSM8k dataset. We see that standard Best-of- N sampling (blue, red, and gold) suffers from reward hacking, exhibiting the characteristic rise-and-fall pattern as N increases. In contrast, caution (our approach, green) consistently improves with larger N , effectively mitigating reward hacking.

to *reward hacking*, the process by which BoN selects responses that exploit \hat{r} as opposed to the ground truth reward r^* we actually care about; while \hat{r} may be a reasonable approximation to r^* on ‘typical’ generations, as N increases, BoN selects more atypical responses lying outside the training distribution of \hat{r} that achieve high \hat{r} scores through spurious correlations rather than genuine quality improvements. In other words, reward hacking is a manifestation of Goodhart’s Law: whenever a metric (in this case \hat{r}) becomes an optimization target, it ceases to be a reliable measure of quality (Goodhart, 1984; Weng, 2024); for example, reward models are known to favor certain formatting preferences and surface-level patterns learned during training that do not necessarily correspond to improved reasoning or correctness (Liu et al., 2024b; Yu et al., 2025; Bukharin et al., 2025).

With the intuition that reward hacking is caused by out-of-distribution (OOD) generations from the perspective of the reward model, several works have proposed mitigation attempts by either improving the reward model (Liu et al., 2025; Yu et al., 2025) or by regularizing the selection process to favor in-distribution responses (Huang et al., 2025b). Unfortunately, the former approach is fundamentally limited by the asymmetric difficulty of the problem: while it is relatively straightforward to obtain representative examples of high-quality responses through careful curation and human annotation, exhaustively characterizing all possible reward hacking strategies is intractable. Representative of the latter work is Huang et al. (2025b), who propose to regularize the selection process to ensure that the distribution of selected responses does not drift too far from the distribution of responses seen during reward model training in a strong information theoretic sense. The authors provide strong theoretical guarantees on the efficacy, monotonicity, and optimality of their approach, and demonstrate that it can be efficiently implemented in practice through a simple rejection sampling scheme (Block & Polyanskiy, 2023). However, this approach is overly conservative in practice, preventing the selection of genuinely better responses that are slightly out-of-distribution, and thus fails to fully leverage the benefits of inference-time scaling. The problem arises because the regularization is *distributional* and thus simultaneously ensures all possible OOD responses are penalized equally, regardless of whether or not they are likely to arise from imperfections in \hat{r} . The starting point for this paper is thus to ask: **Can we design a BoN sampling scheme that is both robust to reward hacking and still able to leverage the full benefits of inference-time scaling?**

Contributions. We answer this question in the affirmative by introducing *caution*, an inference-time instantiation of the *pessimism* principle from Reinforcement Learning (RL) (Jin et al., 2021; Guo et al., 2022). Pessimism relies on lower confidence bounds to avoid selecting OOD actions with uncertain rewards. While Huang et al. (2025b) implements pessimism at the *distribution* level by constraining the sampling distribution to remain close to the base policy, we instead apply it at the *reward* level: we penalize OOD responses by subtracting per-response uncertainty estimates from the scores assigned by \hat{r} , and then select the response with the highest pessimistic score.

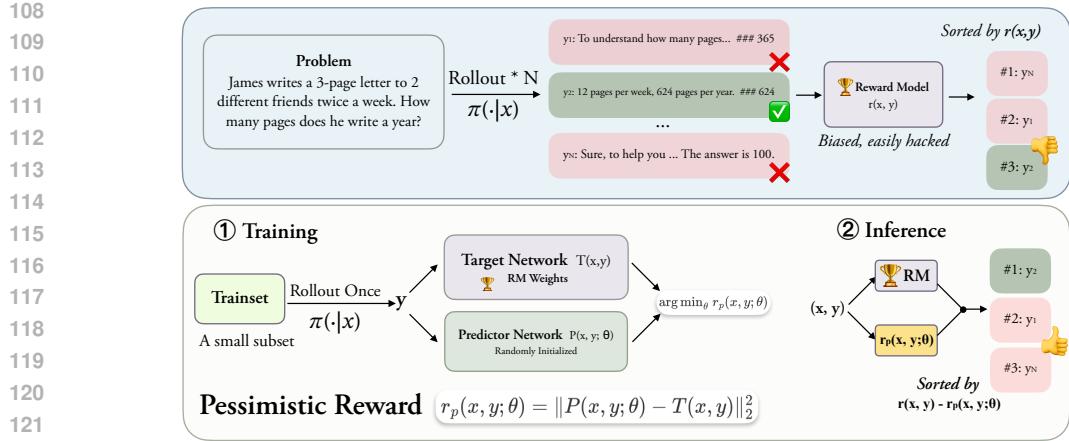


Figure 2: **Overview.** Predictor is trained to match RM features on typical responses; at inference, we select the candidate with the highest pessimistic reward, down-weighting OOD ones.

Conceptually, caution is the dual of *curiosity*, a principle used to drive optimistic exploration in deep RL (Pathak et al., 2017; Burda et al., 2018). As in curiosity, we measure uncertainty by fixing a simple learning target (e.g., a frozen feature embedding), training a student model to predict this target on in-distribution data, and using the prediction error as an uncertainty signal. Unlike curiosity, however, our setting is fully *offline*, so no continual student training is required—making the method significantly simpler and more practical. Through extensive empirical evaluation, we show that caution is computationally efficient and effectively mitigates reward hacking in BoN sampling. We further provide a theoretical analysis in a simplified linear setting, proving that caution strictly improves over standard BoN. Taken together, these results demonstrate that caution is both a powerful practical solution to reward hacking and compelling evidence for the broader efficacy of curiosity-style uncertainty signals in OOD detection and pessimistic policy learning for language models.

2 METHODOLOGY

2.1 PROBLEM FORMULATION

Consider a trained language model (LM) π , where $\pi : \mathcal{X} \rightarrow \Delta(\mathcal{Y})$ is a map from prompts $x \in \mathcal{X}$ to distributions over responses $y \in \mathcal{Y}$. While the LM is typically an autoregressive model, generating one token at a time before feeding the generated tokens back in as context, we abstract this process away and instead consider prompts and responses as single entities, possibly consisting of the concatenation of many individual tokens. Suppose we have access to a learned reward model $\hat{r} : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ that assigns a score to a prompt-response pair (x, y) intended to reflect the quality of the response y to the prompt x . While the learner has access to \hat{r} , the true goal is to output a response that approximately maximizes some ground-truth reward $r^* : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, which is unknown to the learner. We focus on *reward-hacking* behavior that seeks to maximize \hat{r} at the expense of r^* .

Given a prompt x , the learner samples N candidate responses $y_1, \dots, y_N \sim \pi(\cdot|x)$ independently from the LM and then selects one of these responses y_i with $i \in [N]$ to output. Perhaps the simplest strategy is the *Best-of-N* (BoN) strategy, which simply selects the response with the highest reward model score: $\hat{i} = \hat{i}_N = \operatorname{argmax}_{i \in [N]} \hat{r}(x, y_i)$. Unfortunately, this strategy is vulnerable to reward hacking when we only have $\hat{r} \approx r^*$ on typical samples from π . As N grows large, the distribution of \hat{i}_N can differ substantially from that of typical samples from π , leading to poor performance with respect to r^* as shown in (Huang et al., 2025b). While that work proposes to address this issue by constraining the BoN sampling distribution $\hat{\pi}$ to be close to π , we instead propose to directly apply regularization to the reward estimates themselves. Thus, we will design an uncertainty estimate $u : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ that quantifies how uncertain we are about the reward model's estimate $\hat{r}(x, y)$ for a given prompt-response pair (x, y) and then define $\hat{r}_{LCB}(x, y) = \hat{r}(x, y) - \lambda u(x, y)$ for some $\lambda > 0$ to be a pessimistic variant of the reward model that penalizes uncertain responses (Jin et al., 2021). As long as $u(x, y)$ is small for typical samples from $\pi(\cdot|x)$ and large for ‘atypical’ samples,

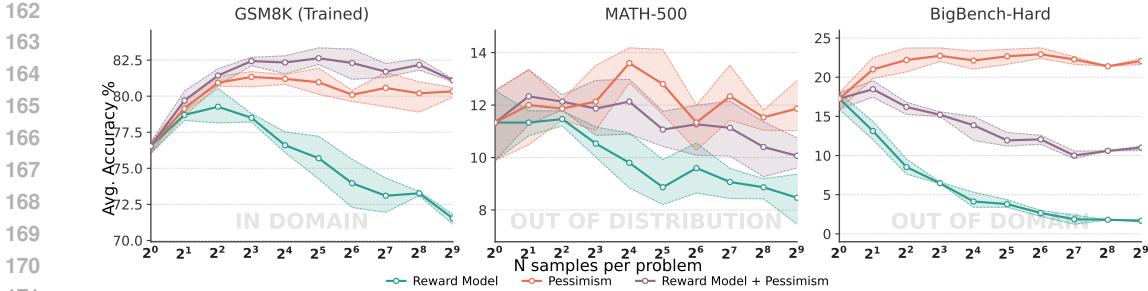


Figure 3: **Scaling over N across distributions and domains.** Best-of- N sampling on GSM8K, MATH-500, and BigBench-Hard. Curves compare selection by Reward Model, Pessimism, and Reward Model + Pessimism. Note that the *pessimism function is trained only on GSM8K*; thus, MATH-500 represents an out-of-distribution setting, while BigBench-Hard represents a fully out-of-domain setting.

this approach will successfully penalize reward-hacking responses by ensuring that $\hat{r}_{LCB} \leq r^*$ and this inequality is approximately tight for actually good quality responses. The final algorithm will thus be to return $\hat{i} = \operatorname{argmax}_{i \in [N]} \hat{r}_{LCB}(x, y_i)$ and the question becomes *what choice of uncertainty estimate u appropriately captures OOD responses in a reward-aware manner?*

2.2 CURIOSITY TO CAUTION: INSTANTIATING PESSIMISM

Our approach introduces the principle of *caution*, a phenomenon dual to the well-known technique of *curiosity* used in online RL to incentivize exploration (Pathak et al., 2017; Burda et al., 2018). While in online RL, OOD states are desirable (as they represent good exploration targets), in our offline setting, OOD responses are to be avoided, as we have no reliable way to estimate their true reward given that $\hat{r} \approx r^*$ only on typical responses from π . We draw inspiration from curiosity (Pathak et al., 2017) and Random Network Distillation (RND) (Burda et al., 2018) in order to ensure that selected responses remain close to the distribution π on which \hat{r} is reliable.

In curiosity and RND, the core idea is to continually train a predictor network to match the outputs of a fixed target that is easily evaluated by the learner; the supervised learning error then becomes a proxy for OOD detection, with the intuition being that the predictor will be accurate on states similar to those seen during training and inaccurate otherwise. While empirically successful as an ‘intrinsic reward’ for optimistic exploration, this method can be challenging to implement due to the continued training of the predictor during online RL, which can lead to nontrivial memory and time overheads in practical RL pipelines. Our method uses the same core idea of using the supervised learning error as a proxy for OOD detection, but is *trained fully offline*, which greatly increases practicality and ease of implementation.

Implementing “Caution” We employ two neural networks operating on the internal representations of the reward model $R(x, y)$: (1) a fixed *target network* $T(x, y) = h_L^R(x, y)$, defined as the hidden state extracted from layer L of the frozen reward model, and (2) a trainable *predictor network* $P_\theta(x, y)$ that learns to predict this hidden state. The first L layers of the reward model serve as our target feature extractor, with their output $h_L^R(x, y)$ as the prediction target. The predictor network $P_\theta(x, y)$ is a separate trainable network with parameters θ that can use various architectural choices (shared embeddings, simplified encoders, projection layers) but is designed to be lightweight to ensure efficient training and inference.

Training Process. We train our predictor P_θ on a dataset $\mathcal{D}_{\text{train}}$ of prompt-response pairs constructed directly from the benchmark train split using mean squared error loss: $\mathcal{L}(\theta) = \mathbb{E}_{(x, y) \sim \mathcal{D}_{\text{train}}} [\|P_\theta(x, y) - T(x, y)\|^2]$, where the norm is Euclidean. We construct responses $y \sim \pi(\cdot | x)$ by sampling from π as an efficient empirical proxy for the training distribution of the reward model. By training on this supervision-free distribution, the predictor achieves low error on in-distribution patterns while preserving high error on distributional outliers. We emphasize that the data requirements for this step are restricted to *prompts*, which are often much cheaper to collect

216
217 **Table 1: Scaling Performance Across Datasets with Reward Hacking Mitigation.** **Peak Acc:**
218 highest accuracy achieved across all N values. **Final Acc:** accuracy at $N = 512$ where reward
219 hacking is most severe. **Degradation:** performance drop from peak to final (lower is better). Results
220 show mean with 95% confidence intervals as subscripts across 3 bootstrap runs. Best results are in
221 **bold** and second-best are underlined.

Method	GSM8K (2021)			MATH-500 (2023)			BBH-Hard (2022)		
	Peak	Final	Degr.	Peak	Final	Degr.	Peak	Final	Degr.
Reward Model	79.3 ± 1.2	71.5 ± 0.3	7.7	11.5 ± 0.3	8.5 ± 1.0	3.0	17.3 ± 1.0	1.7 ± 0.1	15.6
Pessimism Only	81.3 ± 0.5	80.3 ± 0.3	1.0	13.6 ± 0.7	11.9 ± 1.0	1.7	22.9 ± 0.7	22.1 ± 0.4	0.9
RM + Pessimism	82.6 ± 0.6	81.1 ± 0.1	<u>1.5</u>	12.3 ± 1.0	10.1 ± 0.6	<u>2.3</u>	18.5 ± 1.0	11.0 ± 0.3	<u>7.5</u>

222
223 than the high-quality *labelled* data used to train reward models; a gold standard would be to reuse
224 the same prompts used to train the reward model itself, but in practice we find that our method’s
225 OOD detection is relatively robust across different prompt distributions (cf. [Section 3.1](#)).

226
227 **Pessimistic Reward Estimation at Inference Time.** Once we have trained our predictor P_θ , we
228 can use it to define our uncertainty estimate $u(x, y) = \|P_\theta(x, y) - T(x, y)\|^2$, the prediction error
229 of the predictor on the target network’s features. We then plug this into our pessimistic reward
230 estimate $\hat{r}_{LCB}(x, y) = \hat{r}(x, y) - \lambda u(x, y) = \hat{r}(x, y) - \lambda \cdot \|P_\theta(x, y) - T(x, y)\|^2$. Note that this
231 score is very easy to compute at inference time, requiring only two forward passes (one through
232 P_θ and one through T) in addition to the forward pass through the reward model to compute \hat{r} ;
233 note that all of these passes can be fully parallelized and the cost of evaluating $u(x, y)$ is on the
234 same order as that of evaluating \hat{r} due to the reuse of features from \hat{r} . The parameter λ controls the
235 strength of the pessimism penalty, with $\lambda = 0$ recovering standard BoN sampling. The final selection
236 becomes $\hat{i} = \operatorname{argmax}_{i \in [N]} \hat{r}_{LCB}(x, y_i)$, i.e., choosing the response with the highest pessimistic
237 reward estimate. This procedure is summarized in [Figure 2](#).

244 2.3 THEORETICAL INTERPRETATION

245 In order to provide further motivation for our approach, we analyze in [Appendix C](#) a simple theoretical
246 setting in which our approach provably improves upon BoN. While we defer the details to the
247 appendix, we summarize the main theorem here:

248 **Theorem 1** (Informal version of [Theorem 3](#)). *Let $y_1, \dots, y_N \in \mathbb{R}^d$ be i.i.d. samples from a model
249 π and let $r^*(y)$ be a linear reward function. Let $i^* = \operatorname{argmax}_{i \in [N]} r^*(y_i)$ be the optimal response
250 and let $\hat{i} = \operatorname{argmax}_{i \in [N]} \hat{r}(y_i)$ be the response selected by BoN using a learned reward model \hat{r} . If
251 $i_{\text{pess}} = \operatorname{argmax}_{i \in [N]} \hat{r}_{LCB}(y_i)$, where \hat{r}_{LCB} is our caution-regularized reward estimate, then under
252 suitable conditions on the target network, r^* , \hat{r} , π , and the predictor network, it holds that*

$$255 \mathbb{E}[r^*(y_{i_{\text{pess}}}) - r^*(y_i)] \gtrsim \sqrt{\log(N)}, \quad \text{and} \quad \lim_{N \uparrow \infty} \frac{\mathbb{E}[r^*(y_{i^*}) - r^*(y_{i_{\text{pess}}})]}{\mathbb{E}[r^*(y_{i^*})]} = 0.$$

256 While the assumptions and conditions of [Theorem 1](#) are necessarily somewhat strong in order to
257 facilitate the analysis, the theorem provides a proof-of-concept that our approach can provably out-
258 perform BoN in a stylized setting as well as, to the best of our knowledge, the first theoretical
259 guarantee on the success of curiosity- and RND-style methods for OOD detection.

260 3 EXPERIMENTS

261 We now empirically evaluate our proposed approach, with a focus on answering the following three
262 questions: (1) How well does caution mitigate reward hacking as the number of candidates N in-
263 creases? (2) What design decisions (e.g. architecture and training hyperparameters) contribute most
264 to the overall success of our method? (3) In which scenarios does our method outperform standard
265 BoN sampling, and what factors drive its success? We now briefly describe our empirical setup
266 (with full details deferred to [Appendix D](#)) before addressing each of these questions in turn.

270
 271 **Table 2: Weight Ablation and Design Comparison on GSM8K.** We compare different mixing
 272 weights between uncertainty score and \hat{r} score, contrasting our approach (distilling reward model
 273 features) against traditional RND (distilling random network features). **Peak Acc:** best accuracy
 274 across all N . **Final Acc:** accuracy at $N = 512$. Results demonstrate that distilling reward model
 275 representations substantially outperforms random network distillation across weight combinations.

Pessimism strength (λ)	Caution (Ours)		Traditional RND	
	Peak Acc	Final Acc	Peak Acc	Final Acc
0.0 (RM Only)	78.9	71.2	78.9	71.2
0.2	79.5	72.5	78.7	71.3
0.4	80.3	76.0	78.5	72.1
0.6	<u>81.5</u>	<u>80.2</u>	78.4	73.1
0.8	82.1	81.0	78.1	74.9
1.0 (Caution Only)	81.3	79.8	77.0	72.9

286 **Experimental Setup.** We use Llama-3.2-3B-Instruct (Dubey et al., 2024) as our language
 287 model π due to its strong reasoning capabilities and open-source availability.¹ We consider
 288 three prompt distributions coming from reasoning datasets of varying difficulty: GSM8K (Cobbe
 289 et al., 2021), MATH-500 (Hendrycks et al., 2021), and BigBench-Hard (Suzgun et al., 2022).
 290 Our ground truth reward r^* is binary, with reward given if and only if a response provides the
 291 correct answer to the given prompt. While we consider several reward models \hat{r} , we focus
 292 primarily on OASST (Köpf et al., 2023), which Figure 1 demonstrates provides better performance
 293 than comparably sized reward models like Skywork-Reward-V2-Qwen3-0.6B and
 294 Skywork-Reward-V2-Llama-3.2-1B (Liu et al., 2025), making it a strong baseline for demon-
 295 strating reward hacking phenomena. For each prompt and N , we use vLLM (Kwon et al., 2023)
 296 to generate N independent responses from the base model π (and bootstrap this process 3 times to
 297 generate confidence intervals), before scoring each response with \hat{r} and measuring the performance
 298 of the selected response according to r^* . To train our predictor network P_θ , we use an independent
 299 dataset of responses generated by the base model π itself on the training split of GSM8K, and
 300 evaluate the performance of the uncertainty estimates both in-distribution (prompts from the test
 301 set of GSM8K) and out-of-distribution (prompts from other reasoning datasets). In order to maintain
 302 consistency, we normalize all rewards \hat{r} to be centred with unit variance using an independent set
 303 of responses and we do the same for the uncertainty estimates from P_θ .

3.1 CAUTION MITIGATES REWARD HACKING

306 Our main results demonstrate that our proposed mechanism of caution **mitigates reward-hacking**
 307 and **leads to improved performance of BoN sampling as N increases both for in- and out-**
 308 **of-distribution prompts.** In Figure 1, we exhibit the performance of BoN sampling on GSM8K for
 309 several choices of reward models and observe that they all exhibit reward hacking, with performance
 310 degrading for large N . We also see that using our pessimistic approach, this reward-hacking is
 311 substantially mitigated, with performance improving monotonically with N for all considered N .
 312 Moreover, we get a substantial performance boost of 4.2% over peak accuracy for the reward model
 313 alone and an astonishing 15.5% boost over the final accuracy of the reward model alone. This trend
 314 compares favorably with that of Huang et al. (2025b), who observe monotonicity of performance in
 315 N , but struggle to outperform BoN sampling for optimally tuned N for most choices of π and \hat{r} .

316 The above results involve evaluating caution on the same distribution of prompts used to train the
 317 predictor network P_θ , which is likely beneficial to its performance. In reality, we expect to use
 318 caution in scenarios where the prompts are potentially OOD and hope that the uncertainty estimates
 319 remain valid. To evaluate the extent to which this holds, we used the same uncertainty estimates to
 320 produce pessimistic BoN sampling on two significantly harder reasoning datasets: MATH-500 and
 321 BigBench-Hard. These datasets comprise prompts that are significantly different from those in
 322 GSM8K, with the latter even coming from a different, non-mathematical domain.

323 ¹Some other open-weight models are thought to be contaminated with benchmark datasets (Wu et al., 2025),
 which could skew our results, further motivating our choice in model.

324
 325 Table 3: Ablation study of predictor architecture. **Training Loss** measures how well the predictor
 326 fits reward model representations. **Peak Acc** shows best performance across all N . **Final Acc** shows
 327 performance at largest N , where reward hacking is most severe.

Predictor Configuration	Reconstruction Loss	Peak Acc (%)	Final Acc (%)
<i>Simplified Architecture Variants</i>			
Lightweight + Trainable Emb.	0.242	82.2	82.2
Lightweight + Frozen Emb.	0.246	81.5	81.3
Lightweight + Separate Emb.	0.255	82.7	81.8
Lightweight w/o Projection	0.281	81.8	81.3
<i>Full Architecture Variants</i>			
Full + Trainable Emb.	0.127	80.3	80.2
Full + Frozen Emb.	0.127	80.4	78.4

334
 335 We report the results of these OOD prompt experiments in [Figure 3](#) and [Table 1](#), comparing the
 336 performance of the in-distribution prompts (left) with the two other tasks (centre and right). As in
 337 [Figure 1](#), we sweep over a logarithmic grid in N from 1 to 512 and compare three approaches: BoN
 338 using only \hat{r} , BoN using only the uncertainty estimates from P_θ , and BoN using caution. As in
 339 [Figure 1](#), we see that caution successfully mitigates reward hacking while preserving the benefits of
 340 reward-guided selection. The results on MATH-500 reveal a different pattern: pessimism-only out-
 341 performs the combined approach, achieving 13.6% peak accuracy with only 1.7 points degradation
 342 compared to 2.3 points for the combined method. This counterintuitive finding reflects the increased
 343 problem difficulty—while MATH-500 shares the mathematical reasoning domain with GSM8K, the
 344 problems are significantly more complex and multi-step, making the task of \hat{r} more difficult. In this
 345 regime, \hat{r} becomes less reliable at distinguishing genuine quality improvements, making its contribu-
 346 tion less beneficial. However, caution still effectively prevents the severe degradation seen with BoN
 347 (3.0 points). The most challenging task we consider is BigBench-Hard, where the reward model
 348 \hat{r} fails catastrophically. Indeed, we observe the counterintuitive phenomenon that using the uncer-
 349 tainty penalty alone is the most performant of all three methods, doing even better than combining
 350 uncertainty with \hat{r} . This surprising result is due to the fact that when reward models encounter prob-
 351 lems substantially harder than their training distribution, their learned spurious correlations dominate
 352 their judgments, leading to systematic selection of responses that superficially mimic quality patterns
 353 without genuine correctness. Caution, while also operating outside its training distribution, avoids
 354 being misled by these hacking features and maintains stable performance by preferring responses
 355 that match familiar distributional patterns rather than chasing unreliable reward signals.

3.2 ABLATION STUDIES

361 In this section, we dissect the design and implementation of our proposed caution mechanism to
 362 understand which components contribute most to its success. Further ablation studies, including
 363 detailed inspections of uncertainty scores for individual responses and the effect they have on \hat{r}
 364 and r^* scores can be found in [Appendix F](#), while additional details on the ablations can be found
 365 in [Appendix D](#). We evaluate three critical design decisions: (1) the precise architectural choice of
 366 the predictor network; (2) the use of pre-trained embeddings in the target network; (3) the use of a
 367 projector layer between the predictor and target networks. We also conduct two additional ablations
 368 to validate our core design choices: (a) the choice of mixing weights between \hat{r} and the uncertainty
 369 estimates; (b) the choice of distilling reward model features versus random network features.

370 **Architectural Ablations.** We first examine the extent to which the complexity of P_θ and its re-
 371 lationship to the target network T affects performance, in particular whether P_θ needs to share an
 372 architecture with the teacher T . The results are shown in [Table 3](#): while greater flexibility in P_θ leads
 373 to a better reconstruction loss, this does not transfer to better performance when used to instantiate
 374 caution. This occurs because improving the reconstruction loss can actually hurt the ability of P_θ
 375 to detect distributional novelty, as it becomes too good at reconstructing even out-of-distribution
 376 features. We observe a similar trend when considering whether or not to tune the embeddings or
 377 share them between P_θ and T : while sharing embeddings leads to better reconstruction, it can hurt
 the final performance, which suggests that forcing predictors to learn representations from scratch

378	System Prompt: Solve the following problem step by step. Give your final numerical answer at the end with: ##### {NUM}	
379	Question: If Raymond does half as much laundry as Sarah, and Sarah does 4 times as much laundry as David, calculate the difference in the amount of laundry Raymond and David do if Sarah does 400 pounds of laundry.	Question: A pet shop had 6 cages of rodents. 3 of the cages have 10 hamsters and the other 3 have 5 guinea pigs. How many rodents did the pet shop have in total?
380	Response: Let's break down the information given: 1. Sarah does 4 times as much laundry as David. If we represent David's laundry as x , then Sarah's laundry is $4x$. 2. Sarah does 400 pounds of laundry. We can set up the equation $4x = 400$ and solve for x . $4x = 400 \rightarrow x = 400 / 4 \rightarrow x = 100$. So, David does 100 pounds of laundry. 3. Raymond does half as much laundry as Sarah. If we represent Raymond's laundry as y , then $y = (1/2)x$. $y = (1/2) * 100 \rightarrow y = 50$. So, Raymond does 50 pounds of laundry. 4. The difference in the amount of laundry Raymond and David do is: Difference = Raymond's laundry - David's laundry = $50 - 100 = -50$. Since we're looking for the difference, we take the absolute value of the result. The difference in the amount of laundry Raymond and David do is 50 pounds .	Response: To find the total number of rodents, we need to find the total number of hamsters and the total number of guinea pigs. The total number of hamsters is $3 * 10 = 30$. The total number of guinea pigs is $3 * 5 = 15$. Now, we add the number of hamsters and guinea pigs to find the total number of rodents: Total rodents = $30 + 15$. Total rodents = 45
381	Answer: 100 	Answer: 45 
382	Reward: 0.15 (#1 / 1000); Reward + Pessimism: -0.62 (#996 / 1000)	Reward: 0.31 (#976/1000); Reward + Pessimism: 1.72 (# 1 / 1000)
383		
384		
385		
386		

387
388 **Figure 4: Contrasting Selection Behaviors: Reward Hacking vs. Format Compliance.** Two rep-
389 resentative examples showing how reward models favor verbose responses regardless of correctness,
390 while our curiosity-driven pessimism prioritizes format compliance and distributional familiarity.
391 RM assigns high scores to detailed responses regardless of correctness, while pessimism detects
392 distributional deviation from training patterns and prefers correctly formatted solutions.

393
394 creates more sensitive distributional boundaries than inheriting potentially overly simplified features
395 from pre-trained models. Finally, we see that adding a projection layer between the predictor and
396 target networks provides consistent but modest benefits, indicating that information bottlenecks can
397 help prevent pure memorization while preserving signal quality. Together, these findings establish
398 a key empirical finding for instantiating caution: **effective detection requires a careful balance**
399 **between reconstruction accuracy and novelty sensitivity.**

400
401 **Strength of regularization with Caution.** To validate our core design choices, we systematically
402 vary the mixing weights between the uncertainty score and that of \hat{r} to identify the optimal balance
403 for reward hacking mitigation and understand the robustness of our results to this choice. Note
404 that due to the normalization of \hat{r} and uncertainty estimates described above, the strength λ can
405 be viewed as a direct measurement of the influence of the pessimism relative to \hat{r} . We report our
406 findings in [Table 2](#) and [Appendix F](#). We observe that our approach is relatively robust to the choice
407 of λ , with moderate to high weights (0.6-0.8) achieving optimal performance. While the precise
408 optimal weight will vary by task and choice of \hat{r} , with larger λ being required in situations where \hat{r}
409 is less reliable, the results suggest that our approach does not require delicate tuning to be effective.

410
411 **Comparing Caution to RND.** Finally, we compare our approach of distilling reward model fea-
412 tures (motivated by curiosity in [Pathak et al. \(2017\)](#)) against the RND ([Burda et al., 2018](#)), which
413 takes the teacher T to be a randomly initialized network (possibly on top of pre-trained embeddings),
414 testing whether our hypothesis about distributional familiarity requires semantic grounding in the re-
415 ward model’s representations. We again consider a range of mixing weights λ and report the results
416 in [Table 2](#). The results reveal a stark contrast between caution and RND, with the latter exhibiting
417 dramatically inferior performance across all choices of λ . This observation provides evidence for
418 the hypothesis that effective distributional regularization requires semantic grounding: randomly
419 initialized features cannot provide meaningful distributional boundaries, while reward model fea-
420 tures capture task-relevant patterns that enable robust novelty detection. The results establish that
421 curiosity-driven pessimism succeeds not merely because of prediction error signals, but specifically
422 because these signals are computed relative to the reward model’s learned task representations.

423 3.3 WHY IT WORKS: A CASE STUDY

424
425 We now turn to a case study to illustrate the mechanism by which caution mitigates reward hacking
426 and improves performance. In [Figure 4](#), we present two representative examples of questions-
427 response pairs. On the left, we see an instance that \hat{r} scores highly (99.8th percentile of all scored
428 responses) despite being incorrect; this response is verbose and contains multiple mathematical rea-
429 soning steps, but ultimately fails to provide the correct answer and does not follow the required
430 formatting, instead demonstrating exactly the type of reward hacking our method targets: super-
431 ficial mimicry of quality patterns (detailed explanations, step-by-step structure) without genuine
correctness or adherence to specified requirements. Note that the caution score for this response

432 appropriately identifies it as OOD, likely due to its failure to follow the formatting requirements that
 433 are common in high-quality responses in the training data.

434 By contrast, on the right of [Figure 4](#), we see a concise, accurate response following the formatting
 435 requirements. While \hat{r} does not rank this response as well as the first, the pessimism recognizes it as
 436 familiar, matching the patterns of high-quality responses in the training data that consistently follow
 437 formatting specifications and provide direct, correct solutions. This juxtaposition reveals that reward
 438 models can conflate verbosity with quality, particularly at larger N where more elaborate responses
 439 become available. Our method effectively distinguishes between responses that *appear* sophisticated
 440 (verbose explanations) and those that *are* actually correct and compliant with task specifications.

442 4 DISCUSSION

443 In this work, we investigated instantiating pessimistic reward estimation with the principle of *cau-*
 444 *tion*, an approach adapted from curiosity-driven exploration in RL that uses a supervised learning
 445 error as a measure of distributional uncertainty and penalizes the estimated reward of uncertain,
 446 out-of-distribution (OOD) inputs. While our focus was on mitigating reward hacking in Best-of- N
 447 sampling, a particular inference-time scaling technique, our results demonstrate that, when properly
 448 applied on top of pre-trained features, caution can effectively detect OOD samples and instantiate
 449 pessimistic policies in general. While we defer a more complete survey of related work to [Ap-](#)
 450 [pendix B](#), we now provide a brief summary thereof as well as discuss future directions.

451 **Related Work.** Best-of- N sampling was introduced in [Stiennon et al. \(2020\)](#) and has been empirically
 452 investigated in many reasoning settings ([Cobbe et al., 2021](#); [Lightman et al., 2023](#); [Li et al.,](#)
 453 [2022](#); [Brown et al., 2024](#)). While often effective, when the scoring rule is learned (as opposed to
 454 oracular), the approach has long been shown to be vulnerable to reward-hacking ([Amodei et al.,](#)
 455 [2016](#); [Skalse et al., 2022](#); [Gao et al., 2023](#)). While many attempted mitigations of reward-hacking
 456 have been explored for RL *finetuning*, relatively few works have focused on the *inference-time set-*
 457 *ting*. Of particular note is [Huang et al. \(2025b\)](#), which proposes a distributional regularization
 458 approach that samples according to a χ^2 -regularized BoN procedure. While theoretically well-
 459 motivated and empirically effective at ensuring monotonicity in N , the approach is overly conser-
 460 *461* vative in practice. Another approach is that of [Jinnai et al. \(2024\)](#), who apply distributional regulari-
 462 *463* zation with respect to a Wasserstein distance on some embedding space; while potentially effective,
 464 the computation required grows quadratically in N , making it impractical for even moderate values
 465 of N . In contradistinction to these works, our approach applies pessimism directly to the estimated
 466 rewards in a way that naturally leverages the beyond-worst-case errors present in estimated reward
 467 models in a way that is impossible for these distributional regularization approaches.

468 Our key technique of *caution* is built on top of the foundational curiosity ([Pathak et al., 2017](#)) and
 469 Random Network Distillation (RND) ([Burda et al., 2018](#)) from classical deep RL. While the such
 470 techniques have been very popular in aiding exploration, the extent to which they can be used for
 471 OOD detection and pessimistic learning has been a matter of some debate, with [Rezaifar et al.](#)
 472 [\(2022\)](#) claiming negative results and [Ciosek et al. \(2019\)](#); [Nikulin et al. \(2023\)](#) demonstrating some
 473 positive results. While these works are (i) in classical RL or supervised learning settings and (ii) do
 474 not use pre-trained features, our results morally align with the latter camp, demonstrating that such
 475 approaches can be effective for OOD detection and pessimistic learning in language models.

476 **Future Directions.** Our work provides strong evidence that caution, when correctly applied on top
 477 of pre-trained features, can be an effective detector of OOD text. While we instantiate this approach
 478 for pessimistic reward estimation, it is natural to wonder if curiosity can be used as an explicit reward
 479 signal to encourage exploration of novel behaviors either purely during inference or during RL post-
 480 training of reasoning models. While some preliminary work like [Gao et al. \(2025\)](#) has explored this
 481 idea, we believe that their mixed results stem from the fact that the curiosity module was trained
 482 from scratch rather than on top of pre-trained features, which we find to be so effective for OOD
 483 detection. Another interesting direction is to explore the extent to which our proposed approach
 484 can help ensure continued AI alignment in the face of adversarial prompting or distributional shift
 485 due to inference-time scaling. While the tasks we consider in this work are related to reasoning,
 486 we expect that our results would carry over to safety and alignment tasks *mutatis mutandis*, which
 487 would represent a promising new approach to ensuring robust alignment of language models.

486 ETHICS STATEMENT
487488 We affirm adherence to the ICLR Code of Ethics. This work only involve publicly available bench-
489 mark datasets under their respective licenses; we do not collect new human-subject data and process
490 no personally identifiable information.
491492 REPRODUCIBILITY STATEMENT
493494 To ensure reproducibility and transparency of the results within this paper, we document relevant
495 hyperparameters and implementation details in the appendix. We will open-source our codebase
496 along with detailed instructions and scripts.
497498 REFERENCES
499500 Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar,
501 Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 techni-
502 cal report. *arXiv preprint arXiv:2412.08905*, 2024.503 Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. Variational best-of-n alignment. *arXiv*
504 preprint *arXiv:2407.06057*, 2024.505 Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Con-
506 crete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.507 Joshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel*
508 *Machines*. MIT Press, 2007.509 Adam Block and Yury Polyanskiy. The sample complexity of approximate rejection sampling with
510 applications to smoothed online learning. In *The Thirty Sixth Annual Conference on Learning*
511 *Theory*, pp. 228–273. PMLR, 2023.512 Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and
513 Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling.
514 *arXiv preprint arXiv:2407.21787*, 2024.515 Alexander Bukharin, Haifeng Qian, Shengyang Sun, Adithya Renduchintala, Soumye Singhal,
516 Zhilin Wang, Oleksii Kuchaiev, Olivier Delalleau, and Tuo Zhao. Adversarial training of reward
517 models. *arXiv preprint arXiv:2504.06141*, 2025.518 Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network
519 distillation. *arXiv preprint arXiv:1810.12894*, 2018.520 Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémie Scheurer, Javier
521 Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems
522 and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint*
523 *arXiv:2307.15217*, 2023.524 Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep
525 reinforcement learning from human preferences. *Advances in neural information processing sys-*
526 *tems*, 30, 2017.527 Kamil Ciosek, Vincent Fortuin, Ryota Tomioka, Katja Hofmann, and Richard Turner. Conserva-
528 tive uncertainty estimation by fitting prior networks. In *International Conference on Learning*
529 *Representations*, 2019.530 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
531 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to
532 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.533 Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help
534 mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

540 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
 541 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
 542 *arXiv e-prints*, pp. arXiv–2407, 2024.

543 Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham,
 544 Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herd-
 545 ing? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint*
 546 *arXiv:2312.09244*, 2023.

547 Jingtong Gao, Ling Pan, Yeting Wang, Rui Zhong, Chi Lu, Qingpeng Cai, Peng Jiang, and Xi-
 548 angyu Zhao. Navigate the unknown: Enhancing llm reasoning with intrinsic motivation guided
 549 exploration. *arXiv preprint arXiv:2505.17621*, 2025.

550 Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In
 551 *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

552 Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1.
 553 MIT Press, 2016.

554 Charles Goodhart. Monetary theory and practice: the uk experience. 1984.

555 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
 556 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
 557 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

558 Kaiyang Guo, Shao Yunfeng, and Yanhui Geng. Model-based offline reinforcement learning with
 559 pessimism-modulated dynamics belief. *Advances in Neural Information Processing Systems*, 35:
 560 449–461, 2022.

561 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 562 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv*
 563 *preprint arXiv:2103.03874*, 2021.

564 Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief
 565 nets. *Neural Computation*, 18:1527–1554, 2006.

566 Sam Houlston, Alizée Pace, Alexander Immer, and Gunnar Rätsch. Uncertainty-penalized direct
 567 preference optimization. *arXiv preprint arXiv:2410.20187*, 2024.

568 Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime:
 569 Variational information maximizing exploration. *Advances in neural information processing sys-
 570 tems*, 29, 2016.

571 Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jor-
 572 dan T Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening
 573 mechanism. In *The Thirteenth International Conference on Learning Representations*, 2025a.

574 Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster.
 575 Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv*
 576 *preprint arXiv:2503.21878*, 2025b.

577 Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang,
 578 Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*,
 579 2024.

580 Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, Kenshi Abe, Mitsuki Sakamoto, and Eiji
 581 Uchibe. Evaluation of best-of-n sampling strategies for language model alignment. *arXiv preprint*
 582 *arXiv:2502.12668*, 2025.

583 Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and gen-
 584 eralization in neural networks. *Advances in neural information processing systems*, 31, 2018.

585 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
 586 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv*
 587 *preprint arXiv:2412.16720*, 2024.

594 Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In
 595 *International conference on machine learning*, pp. 5084–5096. PMLR, 2021.
 596

597 Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. Regularized best-of-n sampling with
 598 minimum bayes risk objective for language model alignment. *arXiv preprint arXiv:2404.01054*,
 599 2024.

600 Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement
 601 learning from human feedback. 2024.

602 Hadi Khalaf, Claudio Mayrink Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio
 603 du Pin Calmon. Inference-time reward hacking in large language models. *arXiv preprint*
 604 *arXiv:2506.19248*, 2025.

605 Andreas Köpf, Yannic Kilcher, Dimitri Von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith
 606 Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant
 607 conversations-democratizing large language model alignment. *Advances in neural information*
 608 *processing systems*, 36:47669–47681, 2023.

609 Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E.
 610 Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model
 611 serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating*
 612 *Systems Principles*, 2023.

613 Cassidy Laidlaw, Shivam Singhal, and Anca Dragan. Correlated proxies: A new definition and
 614 improved mitigation for reward hacking. *arXiv preprint arXiv:2403.03185*, 2024.

615 Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu,
 616 Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward
 617 models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

618 Jiong Li and Pratik Gajane. Curiosity-driven exploration in sparse-reward multi-agent reinforcement
 619 learning. *arXiv preprint arXiv:2302.10825*, 2023.

620 Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittweiser, Rémi Leblond, Tom
 621 Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation
 622 with alphacode. *Science*, 378(6624):1092–1097, 2022.

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

626 Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang
 627 Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint*
 628 *arXiv:2410.18451*, 2024a.

629 Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen,
 630 Fuxiang Zhang, Jiacheng Xu, et al. Skywork-reward-v2: Scaling preference data curation via
 631 human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

632 Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking
 633 reward models of language models with subtlety and style. *arXiv preprint arXiv:2410.16184*,
 634 2024b.

635 Odelia Melamed, Gilad Yehudai, and Gal Vardi. Adversarial examples exist in two-layer relu net-
 636 works for low dimensional linear subspaces. *Advances in Neural Information Processing Systems*,
 637 36:5028–5049, 2023.

638 Antonio Valerio Miceli-Barone, Alexandra Birch, and Rico Sennrich. Distributionally robust recur-
 639 rent decoders with random network distillation. *arXiv preprint arXiv:2110.13229*, 2021.

640 Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christo-
 641 pher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted
 642 question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

648 Alexander Nikulin, Vladislav Kurenkov, Denis Tarasov, and Sergey Kolesnikov. Anti-exploration
 649 by random network distillation. In *International Conference on Machine Learning*, pp. 26228–
 650 26244. PMLR, 2023.

651

652 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
 653 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to fol-
 654 low instructions with human feedback. *Advances in neural information processing systems*, 35:
 655 27730–27744, 2022.

656

657 Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping
 658 and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

659

660 Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration
 661 by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787.
 662 PMLR, 2017.

663

664 Rafael Rafailev, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 665 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances
 666 in Neural Information Processing Systems*, 36:53728–53741, 2023.

667

668 Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in
 669 neural information processing systems*, 20, 2007.

670

671 Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier
 672 Bachem, and Johan Ferret. Warm: On the benefits of weight averaged reward models. *arXiv
 673 preprint arXiv:2401.12187*, 2024.

674

675 Patrik Reizinger and Márton Szemenyei. Attention-based curiosity-driven exploration in deep rein-
 676 forcement learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech
 677 and Signal Processing (ICASSP)*, pp. 3542–3546. IEEE, 2020.

678

679 Shideh Rezaifar, Robert Dadashi, Nino Vieillard, Léonard Hussenot, Olivier Bachem, Olivier
 680 Pietquin, and Matthieu Geist. Offline reinforcement learning as anti-exploration. In *Proceed-
 681 ings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8106–8114, 2022.

682

683 Wenlei Shi and Xing Jin. Heimdall: test-time scaling on the generative verification. *arXiv preprint
 684 arXiv:2504.10337*, 2025.

685

686 Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and character-
 687 izing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

688

689 Mei Song, Andrea Montanari, and P Nguyen. A mean field view of the landscape of two-layers
 690 neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

691

692 Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford,
 693 Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances
 694 in neural information processing systems*, 33:3008–3021, 2020.

695

696 Haoran Sun, Yekun Chai, Shuhuan Wang, Yu Sun, Hua Wu, and Haifeng Wang. Curiosity-driven
 697 reinforcement learning from human feedback. *arXiv preprint arXiv:2501.11463*, 2025.

698

699 Mirac Suzgun, Nathan Scales, Nathanael Schärlí, Sebastian Gehrmann, Yi Tay, Hyung Won Chung,
 700 Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-
 701 bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*,
 702 2022.

703

704 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez,
 705 Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural informa-
 706 tion processing systems*, 30, 2017.

707

708 Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*,
 709 volume 47. Cambridge university press, 2018.

702 Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge university press, 2019.
 703
 704

705 Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong
 706 Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment
 707 with multi-objective rewards. In *ACL*, 2024a.

708 Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences
 709 via multi-objective reward modeling and mixture-of-experts. In *EMNLP*, 2024b.
 710

711 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdh-
 712 ery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models.
 713 *arXiv preprint arXiv:2203.11171*, 2022.

714 Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang,
 715 Rui Xie, Jindong Wang, Xing Xie, et al. Pandalm: An automatic evaluation benchmark for llm
 716 instruction tuning optimization. *arXiv preprint arXiv:2306.05087*, 2023.
 717

718 Lilian Weng. Reward hacking in language models. *lil'log*, 2024. URL <https://lilianweng.github.io/posts/2024-06-19-reward-hacking>.
 719

720 Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and
 721 Jun Zhao. Large language models are better reasoners with self-verification. *arXiv preprint
 722 arXiv:2212.09561*, 2022.

723

724 Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao
 725 Zhou, Huijie Lv, Ming Zhang, et al. Reasoning or memorization? unreliable results of reinforce-
 726 ment learning due to data contamination. *arXiv preprint arXiv:2507.10532*, 2025.

727 Yuzi Yan, Xingzhou Lou, Jialian Li, Yiping Zhang, Jian Xie, Chao Yu, Yu Wang, Dong Yan, and
 728 Yuan Shen. Reward-robust rlhf in llms. *arXiv preprint arXiv:2409.15360*, 2024.
 729

730 Gilad Yehudai and Ohad Shamir. On the power and limitations of random features for understanding
 731 neural networks. *Advances in neural information processing systems*, 32, 2019.

732

733 Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Zhengran Zeng, Wei Ye, Jindong Wang,
 734 Yue Zhang, and Shikun Zhang. Freeeval: A modular framework for trustworthy and efficient
 735 evaluation of large language models. *arXiv preprint arXiv:2404.06003*, 2024a.

736 Zhuohao Yu, Weizheng Gu, Yidong Wang, Xingru Jiang, Zhengran Zeng, Jindong Wang, Wei Ye,
 737 and Shikun Zhang. Reasoning through execution: Unifying process and outcome rewards for
 738 code generation. *arXiv preprint arXiv:2412.15118*, 2024b.

739

740 Zhuohao Yu, Jiali Zeng, Weizheng Gu, Yidong Wang, Jindong Wang, Fandong Meng, Jie Zhou, Yue
 741 Zhang, Shikun Zhang, and Wei Ye. Rewardanything: Generalizable principle-following reward
 742 models. *arXiv preprint arXiv:2506.03637*, 2025.

743

744 Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang.
 745 Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora en-
 746sembles. *arXiv preprint arXiv:2401.00243*, 2023.

747

748 Jianfei Zhang, Jun Bai, Bei Li, Yanmeng Wang, Rumei Li, Chenghua Lin, and Wenge Rong. Dis-
 749 entangling preference representation and text generation for efficient individual preference align-
 750 ment. *arXiv preprint arXiv:2412.20834*, 2024a.

751

752 Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. Overcoming re-
 753 ward overoptimization via adversarial policy optimization with lightweight uncertainty estima-
 754 tion. *arXiv preprint arXiv:2403.05171*, 2024b.

755 Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul
 756 Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv
 757 preprint arXiv:1909.08593*, 2019.

756 A THE USE OF LARGE LANGUAGE MODELS (LLMs)
757758 LLMs were used assist refining the writing of this paper, including grammar correction, wording
759 refinement, and formatting adjustments. We also use LLM agents to help with finding relevant work
760 and implementing parts of our code. The use of AI tools does not affect the originality of the work
761 or the authors' responsibility for the content.
762763 B ADDITIONAL RELATED WORK
764765 We now give a more detailed discussion of how the present paper relates to prior work. Our approach
766 is situated at the intersection of three key areas of research: Best-of- N sampling with reward models,
767 reward hacking and its mitigation attempts, and curiosity-driven exploration techniques.
768769 **Best-of- N Sampling and Reward Models.** Best-of- N (BoN) sampling generates N can-
770 didate responses and selects the highest-scoring according to a reward model: $y^* =$
771 $\arg \max_{y \in \mathcal{Y}_N} r(x, y)$ (Stiennon et al., 2020). This technique has proven effective for mathemati-
772 cal reasoning (Cobbe et al., 2021; Lightman et al., 2023) and competitive programming (Li et al.,
773 2022) and the version of BoN with oracular rewards (called ‘pass@k’) is a standard evaluation
774 metric for reasoning models (Dubey et al., 2024; Ouyang et al., 2022; Abdin et al., 2024; Hui
775 et al., 2024; Lambert et al., 2024). Reward models are typically trained on tuples of preference
776 data $(x, y_{\text{chosen}}, y_{\text{rejected}})$ consisting of a prompt x as well as preferred and dispreferred responses
777 y_{chosen} and y_{rejected} using ranking losses. While a powerful paradigm, this training procedure intro-
778 duces vulnerabilities: models must infer complex reasons for preferences, often leading them to
779 rely on spurious correlations like response length, formatting patterns, or stylistic preferences (Liu
780 et al., 2024b). High-quality reward models are also computationally expensive to train, requiring re-
781 sources similar to base language models. This cost makes *ensemble approaches*, a common strategy
782 for reducing vulnerabilities in ML systems, impractical, as organizations typically train only a single
783 reward model per domain (Gao et al., 2023; Jinnai et al., 2024). Our work focuses on mitigating
784 reward hacking in BoN sampling with a single reward model, without retraining or modifying the
785 reward model itself.786 **Reward Hacking and Mitigation Attempts.** Reward hacking occurs when optimizing against
787 imperfect proxy rewards leads to high-scoring but low-quality outcomes—a manifestation of Good-
788 hart’s Law (Amodei et al., 2016; Skalse et al., 2022). Gao et al. (2023) observed a common trend in
789 BoN sampling, where as N increases, performance first rises then falls. As demonstrated theoreti-
790 cally in Huang et al. (2025b), these two phases correspond to an initial phase (when N is small) and the
791 learned reward \hat{r} is an effective proxy for the true reward r^* , leading to BoN succeeding, and a
792 second phase where N grows so large so as to produce atypical responses on which \hat{r} is no longer
793 effective due to spurious correlations and poor coverage during training (Eisenstein et al., 2023; Liu
794 et al., 2024b; Yu et al., 2025). Broadly, there have been two main approaches to mitigating this
795 reward-hacking problem: *training-time* and *inference-time* approaches.796 *Training-time ensemble methods* attempt to mitigate this by combining multiple reward
797 models (Coste et al., 2023; Zhai et al., 2023; Ramé et al., 2024; Yan et al., 2024). However, Eisenstein
798 et al. (2023) demonstrate that ensembles reduce but do not eliminate reward hacking. More criti-
799 cally, ensemble methods require multiple expensive reward sources, creating prohibitive costs for
800 practical applications.801 *Inference-time approaches* work with existing reward models and intervene in the sampling pro-
802 cedure itself. Of note, Jinnai et al. (2024) proposed an approach that involves regularizing the BoN
803 selection with respect to a Wasserstein distance in some embedding space and demonstrated em-
804 pirical improvement over BoN as N grows. Unfortunately, this approach requires computation to
805 scale quadratically in N , making it somewhat impractical for larger N . Of greatest relevance to the
806 present work is that of Huang et al. (2025b), who propose instead to use an information-theoretic
807 divergence, the χ^2 -divergence, to regularize the BoN selection. They demonstrate that this approach
808 is statistically and computationally optimal under the assumption that the learned reward model is
809 close to the true reward in expected squared error loss for responses sampled from the base model.
810 While theoretically well-motivated and empirically effective at ensuring monotonicity in N , the
811 approach is overly conservative in practice, leading to suboptimal performance when N is moderate.

810 Another inference-time approach is that of Khalaf et al. (2025), who consider a KL-regularized BoN
 811 procedure as well as a computationally efficient approximation using the Poisson distribution. Un-
 812 fortunately, their method exhibits limited improvement over standard BoN, likely due to the fact
 813 that BoN is already effectively KL-regularized unless N is growing exponentially. Unlike these
 814 works, our approach does not seek to provide *distributional* regularization but instead instantiates
 815 pessimism through the *reward estimates* themselves. Thus, our approach is able to leverage the fact
 816 that reward models may be imperfect in ways that are not information-theoretically ‘worst-case’ and
 817 adapt accordingly.

818 **Curiosity-Driven Exploration and Random Network Distillation.** Curiosity-driven exploration
 819 in reinforcement learning uses prediction error to quantify novelty of given states (Pathak et al.,
 820 2017; Sun et al., 2025; Li & Gajane, 2023). The Intrinsic Curiosity Module (ICM) (Pathak et al.,
 821 2017) uses prediction error $\|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2$ as a curiosity signal, while Random Network
 822 Distillation (RND) (Burda et al., 2018) employs a simpler approach with two networks: a fixed
 823 target network $T(s)$ and a trainable predictor $P(s)$ that minimizes $\|P(s) - T(s)\|_2^2$. In both cases,
 824 the supervised learning error serves as a proxy for OOD detection, with high error indicating states
 825 that are far outside the distribution (and thus worth exploring from the perspective of online RL).
 826 While these methods were originally developed for exploration, they can instead be instantiated
 827 to prevent a policy from moving to far out of the distribution of observed states. While there has
 828 been some debate as to the efficacy of these approaches for OOD detection and pessimistic learning
 829 (Rezaifar et al., 2022; Ciosek et al., 2019; Nikulin et al., 2023), our results suggest that when
 830 properly implemented, such an approach can be effective in language modeling.

832 C THEORY

834 In this section, we give formal statements for and prove our theoretical results. We begin in [Appendix C.1](#)
 835 for formalizing the setting we consider in which pessimism can help mitigate reward
 836 hacking in Best-of- N sampling. We emphasize that this setting is a simplified abstraction intended
 837 to cleanly showcase the benefits of pessimism and help develop intuition for our approach and is
 838 not intended to represent a realistic model of language or reward modeling. We then continue in
 839 [Appendix C.2](#) to prove general results on the performance of BoN and pessimistic selection in our
 840 setting. We proceed in [Appendix C.3](#) to introduce a simplified model of our proposed caution
 841 approach, motivated by Random Network Distillation (RND) (Burda et al., 2018). We prove in simpli-
 842 fied linear and two-layer ReLU settings that this approach can be used to instantiate pessimism under
 843 an idealized optimization model. Finally, we combine these results in [Appendix C.4](#) to prove our
 844 main theorem that our proposed approach improves over BoN sampling in the model we consider.

846 C.1 FORMAL SETTING

848 In order to formalize the setting we consider, we suppose that a learner is given access to a language
 849 model π that maps a prompt x to a distribution over responses y . We will further identify these
 850 prompt-response pairs with their *embeddings* in some linear feature space \mathbb{R}^d and, for the sake of
 851 simplicity, consider results on a per-prompt basis. Thus we will assume a fixed prompt x and write
 852 π for $\pi(x)$. We will further suppose that there exists a *ground truth reward function* that is linear in
 853 the embedded features, i.e.,

$$854 r^*(y) = \langle \theta^*, y \rangle.$$

855 Such an abstraction is partially justified by the fact that many modern reward models are linear
 856 layers on top of pre-trained Language Models (Lambert et al., 2024; Liu et al., 2024a; Wang et al.,
 857 2024a;b) and we thus simply directly associate the concatenation of prompt and response with its
 858 embedding in the final layer of the LM.

859 The goal of the learning problem is similar to that in Huang et al. (2025b): given access to
 860 $y_1, \dots, y_n \sim \pi$ sampled independently, as well as some imperfect proxy reward function $\hat{r} : \mathbb{R}^d \rightarrow$
 861 \mathbb{R} (e.g., a learned reward model), we wish to select a response $y_{\hat{i}}$ with $\hat{i} \in [n]$ such that $r^*(y_{\hat{i}})$ is
 862 as large as possible. Note that if \hat{r} were in fact a perfect reward model, so that $\hat{r} = r^*$, then the
 863 optimal strategy would be to simply select $y_{\hat{i}}$ with $\hat{i} = \text{argmax}_{i \in [n]} \hat{r}(y_i)$, which is the popular

864 *Best-of- N* (BoN) strategy. However, as discussed in the main text, this strategy can fail when \hat{r} is a
 865 poor approximation to r^* due to reward-hacking.
 866

867 Unlike [Huang et al. \(2025b\)](#), wherein the authors suppose that \hat{r} and r^* are close in expected squared
 868 error under π , we will instead suppose that \hat{r} and r^* agree on a low dimensional subspace of \mathbb{R}^d
 869 but may be arbitrarily different on the orthogonal complement. More precisely, suppose that there
 870 exists a linear subspace $V \subset \mathbb{R}^d$ such that $\dim(V) = k \ll d$ and $r^*(y) = \hat{r}(y)$ for all $y \in V$.
 871 We will further make the assumption that r^* and \hat{r} are linear functions, i.e., $\hat{r}(y) = \langle \hat{\theta}, y \rangle$ and
 872 $r^*(y) = \langle \theta^*, y \rangle$ for some $\hat{\theta}, \theta^* \in \mathbb{R}^d$. To aid our analysis, we will make the simplifying assumption
 873 that $\theta^* \in V$, i.e. $\text{proj}_{V^\perp} \theta^* = 0$, representing the fact that the true reward is a function only of certain
 874 linear features; the error between the learned reward \hat{r} and the groundtruth r^* is thus restricted to
 875 OOD effects, conceptualized as the orthogonal complement V^\perp of V .

876 Finally, in order to make the analysis tractable, we will assume that π is a centred Gaussian distri-
 877 bution with covariance $\Sigma \in \mathbb{R}^{d \times d}$, i.e., $\pi = \mathcal{N}(0, \Sigma)$. Note that Σ may have full rank d and thus π
 878 may have full support on \mathbb{R}^d , which ensures that $\hat{r} \neq r^*$ on V^\perp , the orthogonal complement of V .
 879

880 C.2 PESSIMISTIC SELECTION AND BEST-OF- N SAMPLING

882 In this section we provide two key bounds on the performance of Best-of- N sampling and pes-
 883 simistic selection, which will demonstrate the benefits of pessimism in our setting. Before doing so,
 884 we state and prove a simple bound on the best possible performance that can be achieved by any
 885 selection strategy.

886 **Proposition 1.** *Let $y_1, \dots, y_N \sim \pi$ be independent samples from π . Then it holds that*

$$888 \mathbb{E} \left[\max_{i \in [N]} r^*(y_i) \mid \theta^* \right] = \left\| \Sigma^{1/2} \theta^* \right\| \cdot M_N,$$

890 where $M_N = \mathbb{E} \left[\max_{i \in [N]} Z_i \right]$ and Z_1, \dots, Z_N are i.i.d. $\mathcal{N}(0, 1)$ random variables. Moreover, it
 891 holds that

$$893 \sqrt{2 \log(N)} - o(1) \leq M_N \leq \sqrt{2 \log(N)}.$$

895 *Proof.* The second claim is a classical fact about the maximum of Gaussians ([Wainwright, 2019](#);
 896 [Vershynin, 2018](#)), thus it suffices to prove the first statement. Note that

$$898 r^*(y) = \langle \theta^*, y \rangle \sim \mathcal{N}(0, \left\| \Sigma^{1/2} \theta^* \right\|^2).$$

900 Thus $r^*(y) \stackrel{d}{=} \left\| \Sigma^{1/2} \theta^* \right\| Z$ for $Z \sim \mathcal{N}(0, 1)$. The result then follows by positive homogeneity of the
 901 maximum and expectation. \square

903 We now provide a lower bound on the performance of BoN in the setting we consider

905 **Proposition 2.** *Let $y_1, \dots, y_N \sim \pi$ be independent samples from π and let $\hat{i} = \text{argmax}_{i \in [N]} \hat{r}(y_i)$
 906 for $\hat{r}(y) = \langle \hat{\theta}, y \rangle$. Let $i^* = \text{argmax}_{i \in [N]} r^*(y_i)$ for $r^*(y) = \langle \theta^*, y \rangle$. Then it holds that*

$$908 \mathbb{E} \left[r^*(y_{i^*}) - r^*(y_{\hat{i}}) \mid \hat{\theta}, \theta^* \right] = \mathbb{E} [r^*(y_{i^*})] \left(1 - \frac{1}{\sqrt{1 + \frac{\left\| \Sigma^{1/2} \text{proj}_{V^\perp} \hat{\theta} \right\|^2}{\left\| \Sigma^{1/2} \theta^* \right\|^2}}} \right) \\ 910 \\ 911 \\ 912 \\ 913 \\ 914 \\ 915 \\ 916 \\ 917$$

$$= \left(1 - \frac{1}{\sqrt{1 + \frac{\left\| \Sigma^{1/2} \text{proj}_{V^\perp} \hat{\theta} \right\|^2}{\left\| \Sigma^{1/2} \theta^* \right\|^2}}} \right) \cdot \left\| \Sigma^{1/2} \theta^* \right\| \cdot M_N.$$

918 *Proof.* As we have already computed the expectation of $r^*(y_{i^*})$ in [Proposition 1](#), it suffices to
 919 compute the conditional expectation of $r^*(y_i)$. Note that because, conditioned on $\hat{\theta}$ and θ^* , $\hat{r}(y)$ and
 920 $r^*(y)$ are jointly Gaussian, it holds that
 921

$$\begin{aligned}
 922 \mathbb{E}[r^*(y)|\hat{r}(y)] &= \frac{\text{Cov}(\hat{r}(y), r^*(y))}{\text{Var}(\hat{r}(y))} \cdot \hat{r}(y) \\
 923 &= \frac{\langle \theta^*, \Sigma \hat{\theta} \rangle}{\|\Sigma^{1/2} \hat{\theta}\|^2} \cdot \hat{r}(y) \\
 924 &= \frac{\|\Sigma^{1/2} \theta^*\|^2}{\|\Sigma^{1/2} \hat{\theta}\|^2} \cdot \hat{r}(y) \\
 925 &= \frac{\|\Sigma^{1/2} \theta^*\|^2}{\|\Sigma^{1/2} \text{proj}_V \hat{\theta}\|^2 + \|\Sigma^{1/2} \text{proj}_{V^\perp} \hat{\theta}\|^2} \cdot \hat{r}(y) \\
 926 &= \frac{\|\Sigma^{1/2} \theta^*\|^2}{\|\Sigma^{1/2} \theta^*\|^2 + \|\Sigma^{1/2} \text{proj}_{V^\perp} \hat{\theta}\|^2} \cdot \hat{r}(y).
 \end{aligned}$$

927 By the same computation as in [Proposition 1](#), it holds that
 928

$$929 \mathbb{E}[\hat{r}(y_i)|\theta^*, \hat{\theta}] = \|\Sigma^{1/2} \hat{\theta}\| \cdot M_N = \sqrt{\|\Sigma^{1/2} \theta^*\|^2 + \|\Sigma^{1/2} \text{proj}_{V^\perp} \hat{\theta}\|^2} \cdot M_N.$$

930 The result follows by plugging in and rearranging. \square
 931

932 We now show that with pessimism instantiated correctly, we can strictly improve on the performance
 933 of BoN.
 934

935 **Proposition 3.** *Let $y_1, \dots, y_N \sim \pi$ be independent samples from π and let $\alpha : \mathbb{R}^d \rightarrow \mathbb{R}_+$ be a
 936 function satisfying*

$$937 (1 - c) \|\text{proj}_{V^\perp} y\| - \varepsilon \leq \alpha(y) \leq \|\text{proj}_{V^\perp} y\| + \varepsilon$$

938 for some $c \in (0, 1)$ and $\varepsilon > 0$. Let
 939

$$940 i_{\text{pess}} = \underset{i \in [N]}{\text{argmax}} \hat{r}(y_i) - \lambda \cdot \alpha(y_i), \quad \lambda \geq \frac{\|\text{proj}_{V^\perp} \hat{\theta}\|}{1 - c}.$$

941 Then it holds that
 942

$$943 \mathbb{E}[r^*(y_{i^*}) - r^*(y_{i_{\text{pess}}})] \leq \lambda \left(\sqrt{\frac{2}{\pi} \cdot \text{Tr}(\Sigma \text{proj}_{V^\perp})} + 2\varepsilon \right).$$

944 *Proof.* Let
 945

$$946 \hat{r}_{\text{LCB}}(y) = \hat{r}(y) - \lambda \cdot \alpha(y).$$

947 We claim that with the assumption on λ and α , it holds that $\hat{r}_{\text{LCB}}(y) \leq r^*(y) + \lambda \cdot \varepsilon$ for all $y \in \mathbb{R}^d$.
 948 Indeed, note that
 949

$$\begin{aligned}
 950 r^*(y) - \hat{r}_{\text{LCB}}(y) &= \langle \theta^* - \hat{\theta}, y \rangle + \lambda \cdot \alpha(y) \\
 951 &= \langle \text{proj}_{V^\perp} \hat{\theta}, y \rangle + \lambda \cdot \alpha(y) \\
 952 &\geq -\|\text{proj}_{V^\perp} \hat{\theta}\| \cdot \|\text{proj}_{V^\perp} y\| + \lambda \cdot ((1 - c) \|\text{proj}_{V^\perp} y\| - \varepsilon) \\
 953 &\geq -\lambda \varepsilon,
 \end{aligned}$$

972 where the first inequality is by Cauchy-Schwarz.
 973
 974 Now observe that

$$975 \quad r^*(y_{i^*}) - r^*(y_{i_{\text{pess}}}) = [r^*(y_{i^*}) - \hat{r}_{\text{LCB}}(y_{i^*})] + [\hat{r}_{\text{LCB}}(y_{i^*}) - \hat{r}_{\text{LCB}}(y_{i_{\text{pess}}})] + [\hat{r}_{\text{LCB}}(y_{i_{\text{pess}}}) - r^*(y_{i_{\text{pess}}})] \\ 976 \quad \leq r^*(y_{i^*}) - \hat{r}_{\text{LCB}}(y_{i^*}) + \lambda\varepsilon, \\ 977$$

978 where the second group of terms is non-positive by definition of i_{pess} and the last group of terms is
 979 non-positive by the claim above. To conclude, we observe that

$$980 \quad r^*(y_{i^*}) - \hat{r}_{\text{LCB}}(y_{i^*}) = r^*(y_{i^*}) - \hat{r}(y_{i^*}) + \lambda \cdot \alpha(y_{i^*}) \\ 981 \quad = \langle \text{proj}_{V^\perp} \hat{\theta}, y_{i^*} \rangle + \lambda \cdot (\|\text{proj}_{V^\perp} y_{i^*}\| + \varepsilon). \\ 982$$

983 We now observe that as random variables, i^* is independent of the set $\{\text{proj}_{V^\perp} y_i \mid i \in [N]\}$ by Gaussian
 984 orthogonality and the fact that $\text{proj}_{V^\perp} \theta^* = 0$. Thus it holds first that

$$985 \quad \mathbb{E} \left[\langle \text{proj}_{V^\perp} \hat{\theta}, y_{i^*} \rangle \mid \hat{\theta}, \theta^* \right] = \sum_{i=1}^N \mathbb{E} \left[\langle \text{proj}_{V^\perp} \hat{\theta}, y_i \rangle \mid \hat{\theta}, \theta^*, \{i^* = i\} \right] \cdot \Pr[i^* = i \mid \hat{\theta}, \theta^*] \\ 986 \quad = \sum_{i=1}^N \mathbb{E} \left[\langle \text{proj}_{V^\perp} \hat{\theta}, y_i \rangle \mid \hat{\theta}, \theta^* \right] \cdot \Pr[i^* = i \mid \hat{\theta}, \theta^*] \\ 987 \quad = 0, \\ 988 \\ 989 \\ 990 \\ 991 \\ 992$$

993 where the last equality follows from the fact that y_i is centred. For the second term, observe similarly
 994 that

$$995 \quad \mathbb{E} \left[\|\text{proj}_{V^\perp} y_{i^*}\| \mid \hat{\theta}, \theta^* \right] = \sum_{i=1}^N \mathbb{E} \left[\|\text{proj}_{V^\perp} y_i\| \mid \hat{\theta}, \theta^*, \{i^* = i\} \right] \cdot \Pr[i^* = i \mid \hat{\theta}, \theta^*] \\ 996 \quad = \sum_{i=1}^N \mathbb{E} \left[\|\text{proj}_{V^\perp} y_i\| \mid \hat{\theta}, \theta^* \right] \cdot \Pr[i^* = i \mid \hat{\theta}, \theta^*] \\ 997 \\ 998 \\ 999 \\ 1000 \\ 1001 \\ 1002$$

$$= \mathbb{E} \left[\|\text{proj}_{V^\perp} y_1\| \mid \hat{\theta}, \theta^* \right] = \sqrt{\frac{2}{\pi}} \cdot \left\| \Sigma^{1/2} \text{proj}_{V^\perp} \right\|_{\text{F}}.$$

1003 The result follows by combining the above. \square

1004
 1005 Critically, the bound in [Proposition 3](#) does not depend on N and so, as long as $r^*(y_{i^*})$ grows with
 1006 N , the pessimistic algorithm will eventually outperform BoN. We now show that guarantees on α
 1007 can be obtained through Random Network Distillation.

1008 C.3 ACHIEVING OOD DETECTION WITH CAUTION

1009 Above we isolated the key role that $\|\text{proj}_{V^\perp} y\|$ plays in the failure of the greedy algorithm. While in
 1010 the linear setting, projection to V results in the optimal algorithm, it is impractical in general settings,
 1011 where V is unknown. Here we demonstrate that two simplified models of caution, both inspired by
 1012 Random Network Distillation (RND) ([Burda et al., 2018](#)) can be used to approximate projection
 1013 to V and thus yield a pessimistic algorithm that avoids the failure modes of the greedy algorithm.
 1014 Recall that RND involves training a student network $f_{\hat{w}} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ to predict a teacher network
 1015 $f_{w^*} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ on samples from the the explored distribution, which in this case is supported on
 1016 V . The error $\|f_{\hat{w}}(y) - f_{w^*}(y)\|^2$ is then used as a measure of how far out of distribution y is. We
 1017 will consider two simplified models of RND. The first is a linear model, where $f_W(y) = Wy$ for
 1018 $w \in \mathbb{R}^{m \times d}$. The second is a two-layer ReLU network of the form

$$1021 \quad f(y) = \frac{1}{T} \sum_{\ell=1}^T f_{W_\ell}(y), \\ 1022 \\ 1023$$

1024 where

$$1025 \quad f_W(y) = \frac{1}{m} \cdot U \text{ReLU}(W y), \quad \text{ReLU}(u) = \max\{u, 0\}, \quad (1)$$

1026 and $W = (w_1, \dots, w_m) \in \mathbb{R}^{m \times d}$, $U \in \mathbb{R}^{m \times m}$, and ReLU is applied coordinate wise. In both
 1027 cases, we will suppose that the teacher network is fixed with a randomly initialized parameter and
 1028 the student network is trained to minimize squared error on samples from V . We abstract the mini-
 1029 mization in the following definition.

1030 **Definition 1.** Given a distribution \mathcal{D} on \mathbb{R}^d and an initial parameter W^0 , we say that \widehat{W} is
 1031 trained with *gradient based methods* with samples from \mathcal{D} if $\widehat{W} - W^0$ is in the linear span of
 1032 $\{\nabla_W f_W(y)\}_{y \in \text{supp}(\mathcal{D})}$.

1033 This definition abstracts the precise optimization method used to train the student network, but
 1034 captures the key property that the final parameter is obtained by following gradients of the squared
 1035 error loss on samples from \mathcal{D} . Note that this includes gradient descent and its variants as well as
 1036 more sophisticated approaches involving preconditioning, momentum, and continuous time limits.

1037 We begin with the simpler linear case.

1038 **Proposition 4.** Suppose that $W^*, W_0 \in \mathbb{R}^{m \times d}$ are Gaussian random matrices with independent
 1039 $\mathcal{N}(0, 1/m)$ entries. Let \widehat{W} denote the minimizer of the expected squared error on samples from a
 1040 distribution with support V attained through gradient based methods from samples y supported in
 1041 V initialized at W_0 in the sense of [Definition 1](#). Then with probability at least $1 - \delta$ over the choice
 1042 of W^* and W_0 , it holds that for all $y \in \mathbb{R}^d$,

$$1043 \left(1 - C\sqrt{\frac{k + \log(1/\delta)}{m}}\right)^2 \|\text{proj}_{V^\perp} y\|^2 \leq \|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 \leq \left(1 + C\sqrt{\frac{k + \log(1/\delta)}{m}}\right)^2 \|\text{proj}_{V^\perp} y\|^2.$$

1044 In particular, if as long as $m \gg k$, then $\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2$ is a good approximation to
 1045 $\|\text{proj}_{V^\perp} y\|^2$.

1046 *Proof.* We first observe that $\widehat{W} \text{proj}_{V^\perp} = W_0 \text{proj}_{V^\perp}$ since the training data is supported on V .
 1047 Indeed, the gradient of the squared error loss on a sample $y \in V$ is given by

$$1048 \nabla_W \|W y - W^* y\|^2 = 2(W y - W^* y)y^\top = 2(W - W^*)(\text{proj}_V y)y^\top \text{proj}_V^\top,$$

1049 where we used the fact that $y = \text{proj}_V y$ for all $y \in V$. Thus $\nabla_W \|W y - W^* y\|^2 \text{proj}_{V^\perp} = 0$ for
 1050 all $y \in V$ and the claim follows. Moreover, it is immediate that $\widehat{W} \text{proj}_V = W^* \text{proj}_V$ since the loss
 1051 is minimized at \widehat{W} by strong convexity of the loss function. Thus it holds that

$$1052 \|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 = \|(\widehat{W} - W^*)y\|^2 = \|(\widehat{W} - W^*)\text{proj}_{V^\perp} y\|^2 = \|(W_0 - W^*)\text{proj}_{V^\perp} y\|^2.$$

1053 Letting $Z = W_0 - W^*$, we see that Z is a Gaussian random matrix with independent $\mathcal{N}(0, 2/m)$
 1054 entries. We may now apply standard results on the singular values of Gaussian random matrices
 1055 (cf. e.g. [Vershynin \(2018, Theorem 4.6.1\)](#)) to observe that there is some constant C such that with
 1056 probability at least $1 - \delta$,

$$1057 1 - C\sqrt{\frac{k + \log(1/\delta)}{m}} \leq \lambda_{\min}(Z) \leq \lambda_{\max}(Z) \leq 1 + C\sqrt{\frac{k + \log(1/\delta)}{m}}.$$

1058 This suffices to prove the first statement. The second statement follows immediately. \square

1059 Note that the above result shows that the RND error is a good approximation to $\|\text{proj}_{V^\perp} y\|^2$ uni-
 1060 formly over all $y \in \mathbb{R}^d$ with high probability as long as the embedding dimension is sufficiently
 1061 large relative to the intrinsic dimension of the explored distribution. While this is a nice first step,
 1062 the lack of flexibility of linear functions is a significant limitation. We now proceed to the hidden
 1063 layer ReLU network case.

1064 Instead of assuming that $f_W(y)$ is a linear function, we now suppose that $f_W(y)$ is a two-layer ReLU
 1065 network of the form given in (1). For the sake of simplicity, we assume that only the weights W
 1066 are trained and that the second layer weights $u_i \sim \mathcal{N}(0, 1/m)$ are fixed. Note that, while practically
 1067 unrealistic, the assumption that only a single layer is trained is common in the study of deep learning
 1068 (cf. e.g. [\(Jacot et al., 2018; Yehudai & Shamir, 2019; Song et al., 2018\)](#) and the references therein)

and is motivated by the utility of random features (Rahimi & Recht, 2007). Our analysis is inspired by that of Melamed et al. (2023), who consider a similar model of neural networks, but for the very different aim of investigating adversarial robustness. We show that as long as the number of hidden features m is sufficiently large relative to $\|y\|$, then the RND error is again a good approximation to $\|\text{proj}_{V^\perp} y\|^2$ uniformly over all $y \in \mathbb{R}^d$ with bounded norm with high probability.

Theorem 2. *Let $W^*, W^0 \in \mathbb{R}^{m \times d}$ be Gaussian random matrices with independent $\mathcal{N}(0, 1)$ entries with rows w_i^* and w_i^0 respectively. Let $u_i^*, u_0 \sim \mathcal{N}(0, 1)$ and let f_{W^*} and f_{W^0} be the corresponding ReLU networks as in (1). Suppose (i) that \widehat{W} is obtained through a gradient-based method from a distribution with support V initialized at W^0 as in Definition 1 and (ii) that $f_{\widehat{W}}(y) = f_{W^*}(y)$ for all $y \in V$. Then it holds with probability at least $1 - \delta$ over the choice of W^*, W^0, u_i, u_0 that simultaneously for all $y \in \mathbb{R}^d$,*

$$c \|\text{proj}_{V^\perp} y\|^2 - C \|y\|^2 \sqrt{\frac{d \log(dm/\delta)}{Tm}} - C \|y\|^2 \frac{d \log(dm/\delta)}{Tm} \leq \|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2$$

and

$$\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 \leq \|\text{proj}_{V^\perp} y\|^2 + C \|y\|^2 \sqrt{\frac{d \log(dm/\delta)}{Tm}} + C \|y\|^2 \frac{d \log(dm/\delta)}{Tm},$$

In particular, for $Tm \gg \|y\|^4$ and up to constants, $\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 \asymp \|\text{proj}_{V^\perp} y\|^2$ with high probability.

Proof. We first prove the result for $T = 1$. This step rests four lemmata. We first show in Lemma 1 the key property that, due to the training data being supported on V and the fact that we are only training the first layer weights, it holds that $\widehat{W} \text{proj}_{V^\perp} = W^0 \text{proj}_{V^\perp}$, i.e., weights in the orthogonal complement of V are never changed during training. By the assumption that we have trained to convergence, then, it holds that $W^* \text{proj}_V = \widehat{W} \text{proj}_V$. We then use prove in Lemma 2 a decomposition of the RND error that rests on the precise functional form of the ReLU network. Using this decomposition, we are able to provide bounds on the mean and concentration of the RND error in Lemma 3 and Lemma 4 in the special case that $\|y\| = 1$. The result then follows by combining these lemmata with the positive homogeneity property of ReLU networks: for any $r > 0$, it holds that $f_W(ry) = r f_W(y)$ for all $y \in \mathbb{R}^d$ and all $W \in \mathbb{R}^{m \times d}$.

Now that the result is proved for $T = 1$, the general case follows by tensorization across the independent W_ℓ . \square

Lemma 1. *Let f_{W^*} , f_{W^0} , and $f_{\widehat{W}}$ be as in Theorem 2. Then it holds that $W^0 \text{proj}_{V^\perp} = \widehat{W} \text{proj}_{V^\perp}$.*

Proof. By Definition 1 suffices to show that for any $y \in V$, $(\nabla_W f_W(y)) \text{proj}_{V^\perp} = 0$. To see this, observe that

$$\nabla_W f_W(y)_i = \frac{1}{m} \sum_{j=1}^m u_{ij} \mathbb{I}[\langle w_j, y \rangle > 0] y.$$

Because $y \text{proj}_{V^\perp} = 0$ for all $y \in V$, it follows that $\nabla_W f_W(y) \text{proj}_{V^\perp} = 0$. The result follows. \square

Lemma 2. *Let f_{W^*} , f_{W^0} , and $f_{\widehat{W}}$ be as in Theorem 2. Then with probability at least $1 - \delta$ over the choice of W^*, W^0, u_i, u_0 , it holds that for all $y \in \mathbb{R}^d$,*

$$(f_{\widehat{W}}(y) - f_{W^*}(y))_j = \frac{1}{m} \left(\sum_{i=1}^m \langle u_{ji}^0 w_i^0, \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle \text{proj}_V w_i^* + \text{proj}_{V^\perp} w_i^0, \text{proj}_V y + t \text{proj}_{V^\perp} y \rangle > 0] dt \right) - \frac{1}{m} \left(\sum_{i=1}^m \langle u_{ji}^* w_i^*, \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle w_i^*, \text{proj}_V y + t \text{proj}_{V^\perp} y \rangle > 0] dt \right).$$

1134 *Proof.* By the fundamental theorem of calculus, we have that
 1135

$$1136 \quad m(f_{\widehat{W}}(y) - f_{\widehat{W}}(\text{proj}_V y))_j = \int_0^1 \langle \nabla_y f_{\widehat{W}}(\text{proj}_V y + t(\text{proj}_{V^\perp} y - \text{proj}_V y))_j, y - \text{proj}_V y \rangle dt \\
 1137 \\
 1138 \\
 1139 \quad = \int_0^1 \langle \nabla_y f_{\widehat{W}}(\text{proj}_V y + t\text{proj}_{V^\perp} y)_j, \text{proj}_{V^\perp} y \rangle dt.$$

1140 For any W , it holds that
 1141

$$1142 \quad m \nabla_y f_W(y)_j = \sum_{i=1}^m u_{ji} w_i \mathbb{I}[\langle w_i, y \rangle > 0]. \\
 1143 \\
 1144$$

1145 Thus by the linearity of the integral and [Lemma 1](#), it holds that
 1146

$$1147 \quad m(f_{\widehat{W}}(y) - f_{\widehat{W}}(\text{proj}_V y))_j = \sum_{i=1}^m \langle u_{ji} \hat{w}_i, \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle \hat{w}_i, \text{proj}_V y + t\text{proj}_{V^\perp} y \rangle > 0] dt \\
 1148 \\
 1149 \\
 1150 \quad = \sum_{i=1}^m \langle u_{ji} w_i^0, \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle \text{proj}_V w_i^* + \text{proj}_{V^\perp} w_i^0, \text{proj}_V y + t\text{proj}_{V^\perp} y \rangle > 0] dt,$$

1151 where we used the fact that $\widehat{W} = W^* \text{proj}_V + W^0 \text{proj}_{V^\perp}$ by [Lemma 1](#). A similar, but simpler,
 1152 argument applies to f_{W^*} . We now use the fact that $\widehat{W} \text{proj}_V = W^* \text{proj}_V$ to observe that
 1153

$$1154 \quad f_{\widehat{W}}(y) - f_{W^*}(y) = f_{\widehat{W}}(y) - f_{\widehat{W}}(\text{proj}_V y) - (f_{W^*}(y) - f_{W^*}(\text{proj}_V y))$$

1155 and the result follows. \square
 1156

1157 **Lemma 3.** *Let f_{W^*} , f_{W^0} , and $f_{\widehat{W}}$ be as in Theorem 2. Then it holds that*
 1158

$$1159 \quad \frac{\|\text{proj}_{V^\perp} y\|^2}{4} \leq \mathbb{E} \left[\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 \right] \leq \|\text{proj}_{V^\perp} y\|^2.$$

1160 *Proof.* Let
 1161

$$1162 \quad g(w, y) = \langle w, \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle w, \text{proj}_V y + t\text{proj}_{V^\perp} y \rangle > 0] dt \quad (2)$$

1163 and let
 1164

$$1165 \quad g'(w, w', y) = \langle w', \text{proj}_{V^\perp} y \rangle \int_0^1 \mathbb{I}[\langle \text{proj}_V w + \text{proj}_{V^\perp} w', \text{proj}_V y + t\text{proj}_{V^\perp} y \rangle > 0] dt. \quad (3)$$

1166 By [Lemma 2](#) it holds that
 1167

$$1168 \quad \mathbb{E} \left[(f_{\widehat{W}}(y) - f_{W^*}(y))^2 \right] = \frac{1}{m^2} \mathbb{E} \left[\left(\sum_{i=1}^m u_{ji}^0 g'(w_i^*, w_i^0, y) - u_{ji}^* g(w_i^*, y) \right)^2 \right] \\
 1169 \\
 1170 \quad = \frac{2}{m} \mathbb{E} [g(w_1^*, y)^2],$$

1171 because the u_{ji}^0 and u_{ji}^* are independent and have variance 1 and $g'(w_i^*, w_i^0, y)$ and $g(w_i^*, y)$ are
 1172 independent of u_{ji}^0 and u_{ji}^* and identically distributed. Note now that
 1173

$$1174 \quad g(w_i^*, y)^2 \leq \langle \text{proj}_{V^\perp} y, \text{proj}_{V^\perp} y \rangle^2$$

1175 and thus has expectation at most $\|\text{proj}_{V^\perp} y\|^2$. For the lower bound, observe that
 1176

$$1177 \quad g(w_i^*, y)^2 = \int_0^1 \int_0^1 \langle w_i^*, \text{proj}_{V^\perp} y \rangle^2 \mathbb{I}[\langle w_i^*, \text{proj}_V y + t\text{proj}_{V^\perp} y \rangle > 0] \mathbb{I}[\langle w_i^*, \text{proj}_V y + s\text{proj}_{V^\perp} y \rangle > 0] ds dt \\
 1178 \\
 1179 \geq \int_0^1 \int_0^1 \langle w_i^*, \text{proj}_{V^\perp} y \rangle^2 \mathbb{I}[\langle w_i^*, \text{proj}_V y \rangle > 0] \mathbb{I}[\langle w_i^*, \text{proj}_{V^\perp} y \rangle > 0] ds dt \\
 1180 \\
 1181 \geq \frac{\langle w_i^*, \text{proj}_{V^\perp} y \rangle^2}{4}.$$

1188 Thus

1189
$$\frac{\|\text{proj}_{V^\perp} y\|^2}{4m} = \frac{\langle w_i^*, \text{proj}_{V^\perp} y \rangle^2}{4m} \leq \mathbb{E} [(f_{\widehat{W}}(y) - f_{W^*}(y))_j^2] \leq \frac{\mathbb{E} [\langle w_i^*, \text{proj}_{V^\perp} y \rangle^2]}{m} = \frac{\|\text{proj}_{V^\perp} y\|^2}{m}.$$

1190 Summing over j gives the result. \square

1193 **Lemma 4.** *Let $f_{\widehat{W}}$, f_{W^0} , and f_{W^*} be as in Theorem 2. Then with probability at least $1 - \delta$ over the*

1194 *choice of W^* , W^0 , u_i , u_0 , it holds that uniformly in y for $\|y\| = 1$,*

1196
$$\left| \|\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 - \mathbb{E} [\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2] \right| \leq C \left(\sqrt{\frac{d \log(dm/\delta)}{m}} + \frac{d \log(dm/\delta)}{m} \right).$$

1197

1198 *Proof.* Begin by noting that by positive homogeneity, it suffices to set $\|y\| = 1$. Continuing with
1199 the notation introduced in (2) and (3) we see that, conditional on w_i^*, w_i^0 , it holds that $(f_{\widehat{W}}(y) -$
1200 $f_{W^*}(y))_j$ are independent centred Gaussians with variancse given by
1201

1202
$$\text{Var}((f_{\widehat{W}}(y) - f_{W^*}(y))_j | w_i^*, w_i^0) = \frac{1}{m^2} \sum_{i=1}^m g'(w_i^*, w_i^0, y)^2 + g(w_i^*, y)^2.$$

1203 Thus by standard bounds on the concentration of norm of Gaussian vectors, it holds that with prob-
1204 ability at least $1 - \delta$,

1205
$$\begin{aligned} & \left| \|\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 - \frac{1}{m} \sum_{i=1}^m g'(w_i^*, w_i^0, y)^2 + g(w_i^*, y)^2 \right| \\ & \leq C \sqrt{m \cdot \left(\frac{1}{m^2} \sum_{i=1}^m g'(w_i^*, w_i^0, y)^2 + g(w_i^*, y)^2 \right)^2 \log(1/\delta)} \\ & \quad + C \left(\frac{1}{m^2} \sum_{i=1}^m g'(w_i^*, w_i^0, y)^2 + g(w_i^*, y)^2 \right) \log(1/\delta). \end{aligned}$$

1206 Letting

1207
$$G(w^*, w^0, y) = \frac{1}{m} \sum_{i=1}^m g'(w_i^*, w_i^0, y)^2 + g(w_i^*, y)^2,$$

1208 we see that

1209
$$\left| \|\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 - G(w^*, w^0, y) \right| \leq C \cdot G(w^*, w^0, y) \left(\sqrt{\frac{\log(1/\delta)}{m}} + \frac{\log(1/\delta)}{m} \right). \quad (4)$$

1210 We now demonstrate that $G(w^*, w^0, y)$ concentrates around its mean for fixed y . To do this, we
1211 will first observe that $g(w, y)$ and $g'(w, w', y)$ are identically distributed and thus it suffices to show
1212 concentration for $g(w, y)$ and apply a union bound. Indeed, we have that
1213

1214
$$g(w_i^*, y)^2 \leq \langle w_i^*, \text{proj}_{V^\perp} y \rangle^2$$

1215 and thus has Orlicz ψ_1 norm at most $C \|\text{proj}_{V^\perp} y\|^2$ for some constant C . Thus by standard concen-
1216 tration results for sums of independent subexponential random variables (cf. e.g. [Vershynin \(2018\)](#);
1217 [Wainwright \(2019\)](#)), it holds that with probability at least $1 - \delta$,

1218
$$\left| \frac{1}{m} \sum_{i=1}^m g(w_i^*, y)^2 - \mathbb{E} [g(w_i^*, y)^2] \right| \leq C \|\text{proj}_{V^\perp} y\|^2 \left(\sqrt{\frac{\log(1/\delta)}{m}} + \frac{\log(1/\delta)}{m} \right).$$

1219 Combining this argument with the triangle inequality and (4) along with the fact that $\|y\| = 1$ and
1220 [Lemma 3](#) gives that with probability at least $1 - \delta$,

1221
$$\left| \|\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2 - \mathbb{E} [\|f_{\widehat{W}}(y) - f_{W^*}(y)\|^2] \right| \leq C \left(\sqrt{\frac{\log(1/\delta)}{m}} + \frac{\log(1/\delta)}{m} \right).$$

1222 We now observe that by standard high probability bounds on the operator norms of Gaussian random
1223 matrices (cf. e.g. [Vershynin \(2018, Theorem 4.6.1\)](#)), it holds that with probability at least $1 - \delta$,
1224 that $f_{\widehat{W}}$ and f_{W^*} are C -Lipschitz as is $\mathbb{E} [g(w_i^*, y)^2]$ in y for $\|y\| = 1$. Thus by a standard covering
1225 argument on the unit sphere and a union bound, the result follows. \square

1242 C.4 MAIN RESULT
12431244 We now state our main result, which combines the analysis of pessimism in [Appendix C.2](#) with the
1245 analysis of caution in [Appendix C.3](#).1246 **Theorem 3.** *Let V be a k -dimensional subspace of \mathbb{R}^d and let Σ be a positive semidefinite matrix
1247 such that $\text{Tr}(\Sigma \text{proj}_{V^\perp}) > 0$. Let $\{y_i\}_{i=1}^N$ be i.i.d. samples from $\mathcal{N}(0, \Sigma)$ and let $r^*(y) = \langle \theta^*, y \rangle$
1248 for some $\theta^* \in V$. Suppose that $\hat{\theta} \in \mathbb{R}^d$ such that $\text{proj}_V \hat{\theta} = \theta^*$ and let $\hat{r}(y) = \langle \hat{\theta}, y \rangle$. Let
1249 $\hat{r}_{\text{LCB}}(y) = \hat{r}(y) - \lambda \cdot \alpha(y)$, for $\alpha(y) = \|f_{\hat{W}}(y) - f_W(y)\|$ with $f_W(y)$ being either the linear
1250 model considered in [Proposition 4](#) or the one hidden layer ReLU network considered in [Theorem 2](#).
1251 Let $\hat{i} = \text{argmax}_{i \in [N]} \hat{r}_{\text{LCB}}(y_i)$ and $i^* = \text{argmax}_{i \in [N]} r^*(y_i)$. In the case that f_W is linear, as long
1252 as $m \gtrsim \frac{k(d-k)\|\text{proj}_{V^\perp} \hat{\theta}\|^2}{\log(N)}$ and $\lambda = \Theta(\|\text{proj}_{V^\perp} \hat{\theta}\|)$, it holds that
1253*

1254
$$\mathbb{E}[r^*(y_{i_{\text{pess}}}) - r^*(y_i)] \gtrsim \sqrt{\log(N)}.$$

1255

1256
1257 In the case that f_W is a one hidden layer ReLU network, as long as $Tm \gtrsim \frac{d(d-k)\|\text{proj}_{V^\perp} \hat{\theta}\|^2}{\log(N)}$, the
1258 same holds with $\lambda = \Theta(\|\text{proj}_{V^\perp} \hat{\theta}\|)$. Moreover, in both of these cases it holds that
1259

1260
$$\lim_{N \uparrow \infty} \frac{\mathbb{E}[r^*(y_{i^*}) - r^*(y_{i_{\text{pess}}})]}{\mathbb{E}[r^*(y_{i^*})]} = 0 < \lim_{N \uparrow \infty} \frac{\mathbb{E}[r^*(y_{i^*}) - r^*(y_i)]}{\mathbb{E}[r^*(y_{i^*})]}.$$

1261
1262

1263 *Proof.* The result follows immediately by combining [Proposition 3](#) with [Proposition 4](#) and [Theorem 2](#) and observing that $\|\text{proj}_{V^\perp} y\| \lesssim \sqrt{(d-k)\log(1/\delta)}$ with probability at least $1-\delta$ by standard
1264 concentration results for chi-squared random variables (cf. e.g. [Wainwright \(2019\)](#)). \square
1265
12661267 D EXPERIMENTAL DETAILS
12681269 **Hyperparameter Configuration.** [Table 4](#) provides a comprehensive overview of all hyperparameters
1270 used throughout our experimental evaluation. The configuration represents a careful balance
1271 between computational efficiency and model expressiveness, with key design choices motivated by
1272 our theoretical analysis and empirical ablations.
12731274 The RND architecture employs 10 layers for both target and predictor networks, significantly deeper
1275 than the default 4 layers, which we found provides better representation quality for uncertainty es-
1276 timation. The RND weight $\lambda = 0.2$ represents a moderate pessimism strength that effectively
1277 mitigates reward hacking while preserving the benefits of reward-guided selection. Training hyper-
1278 parameters including the reduced learning rate (1×10^{-5}) and extended training duration (5 epochs)
1279 ensure stable convergence of the predictor network on the GSM8K training distribution.
12801281 **Architectural Ablations.** For the architectural ablation study, we design different levels of net-
1282 work complexity and check their impact on the training objective, the reconstruction loss and also
1283 the final accuracies. [Table 5](#) contains a comprehensive introduction of settings involved in our abla-
1284 tion studies.
12851286 **Implementation Framework.** Our experimental setup utilizes PyTorch 2.3.1 as the primary deep
1287 learning framework, with model inference accelerated through vLLM 0.10.0 and Transformers
1288 4.55.1 for efficient large-scale language model deployment. For RND training, we extract repre-
1289 sentations from the first 10 layers of both predictor and target networks, employing a learning rate
1290 of 1×10^{-5} with 50 warmup steps across 5 training epochs. The RND models are trained on out-
1291 puts generated by the backbone model using the GSM8K training set, and subsequently evaluated
1292 against the validation set of GSM8K as well as Math-500 and BigBench-Hard benchmarks to assess
1293 generalization capabilities.
12941295 **Inference Configuration.** All inference experiments maintain consistent hyperparameters with
1296 temperature set to 1.0 and maximum token limits of 500 for GSM8K and 1024 for Math-500 and
1297 BigBench-Hard evaluations. The increased token limit for harder datasets is necessary because these
1298

1296 benchmarks require significantly more reasoning tokens to avoid truncation before reaching a con-
 1297 clusion. Response generation uses vLLM for efficient parallel sampling across multiple candidates
 1298 in Best-of-N evaluation.

1300 E ADDITIONAL CASE STUDY

1302 To gain deeper insights into the mechanisms underlying reward hacking and our caution-based mit-
 1303 igation, we analyze the correlation patterns between reward model scores and pessimism scores
 1304 through detailed scatter plot visualizations. Each plot displays z-normalized scores for all responses
 1305 to individual GSM8K problems, with green points representing correct responses and red points
 1306 representing incorrect responses.

1308 The scatter plots in [Figure 5](#) reveal two critical failure modes of reward models that our approach
 1309 successfully identifies. In the **high reward, low pessimism region** (upper-left quadrant), we ob-
 1310 serve responses that exemplify systematic reward hacking. These responses achieve high reward
 1311 scores not through genuine correctness or adherence to task requirements, but by exploiting spu-
 1312 rious correlations that the reward model learned during training. Crucially, these responses often
 1313 **ignore fundamental formatting requirements** of the mathematical reasoning task, such as provid-
 1314 ing the final answer in the required format, yet still receive high rewards because the reward model
 1315 prioritizes superficial indicators like verbosity, step-by-step presentation, or mathematical terminol-
 1316 ogy over actual task compliance. This reveals that reward models can be systematically misled by
 1317 responses that mimic the surface patterns of high-quality reasoning without delivering the essential
 1318 components of a correct solution.

1319 Conversely, the **low reward, high pessimism region** (lower-right quadrant) contains responses that
 1320 follow proper formatting conventions and adhere closely to the expected task structure, yet receive
 1321 low reward scores. This pattern illuminates a fundamental limitation of reward models: they func-
 1322 tion primarily as **distributional fitness measures** rather than objective quality assessors. These
 1323 well-formatted responses are penalized not because they lack correctness or clarity, but because
 1324 they deviate from the specific stylistic preferences and response patterns that dominated the re-
 1325 ward model’s training distribution. The reward model essentially measures how closely a response
 1326 matches its learned notion of “preferred” responses rather than evaluating genuine task performance
 1327 or adherence to explicit instructions.

1328 This analysis demonstrates that our caution mechanism successfully identifies both forms of reward
 1329 model failure: it flags spurious high-reward responses that exploit correlational biases while rec-
 1330 ognizing genuinely task-compliant responses that happen to fall outside the reward model’s narrow
 1331 preference distribution. The results underscore that effective reward hacking mitigation requires
 1332 moving beyond simple score-based selection toward distributional awareness that can distinguish
 1333 between genuine quality and superficial pattern matching.

1334 F DETAILED RESULTS FOR ABLATION STUDIES

1335 This section complements [Table 2](#) with full scaling curves. We sweep pessimism
 1336 strength $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and vary the Best-of- N budget over $N \in$
 1337 $\{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$. [Figure 6](#) shows our caution variant that uses reward-model
 1338 features; [Figure 7](#) shows traditional RND with random targets.

1339 **Caution (RM features).** Moderate–high λ (about 0.6–0.8) maintains or improves accuracy as N
 1340 grows, preventing the reward-hacking drop seen at $\lambda = 0$ (RM-only). $\lambda = 1.0$ (pessimism-only) is
 1341 competitive but slightly conservative at small N , this is because for easier problems, most responses
 1342 contain the correct answer but not all of them strictly follows the specified answering format, and
 1343 applying pessimism only would at least filter out the incorrectly formatted responses. Overall, $\lambda \in$
 1344 $[0.6, 0.8]$ delivers the best trade-off across most N .

1345 **Traditional RND (random targets).** Accuracy remains flat or declines with N for all λ , and
 1346 rarely exceeds the RM-only baseline. Sweeping λ offers little benefit, indicating that random targets
 1347 lack the semantic grounding needed for useful uncertainty estimates.

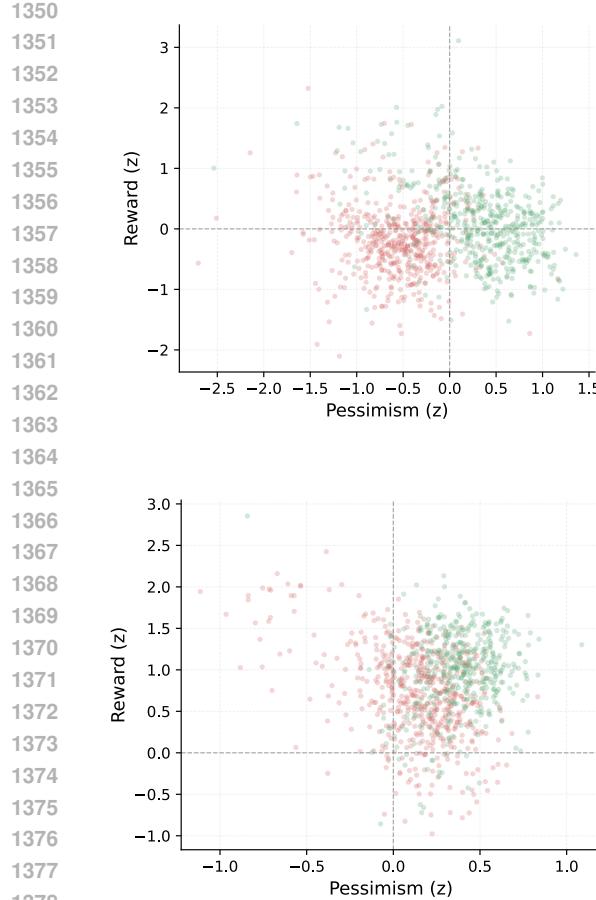


Figure 5: **Pessimism–Reward visualization on GSM8K.** Each row shows one problem: a scatter plot of z-normalized pessimism (x-axis) and z-normalized reward (y-axis), with green points for correct responses and red for incorrect. Upper-left points (high reward, low pessimism) illustrate reward hacking—responses that score well despite low distributional support. Lower-right points (low reward, high pessimism) are well-formed, instruction-following responses that the reward model undervalues; our caution mechanism up-weights these relative to reward-only selection.

Number of Layers (L). In Table 6, we study how the selection of L would impact performance, on a smaller subset of GSM8K with 100 problems and 100 responses for each problem. Generally we observe higher peak accuracies and less degradation when given more layers, although at the cost of compute budget.

G ADDITIONAL BASELINE COMPARISONS

To further validate the effectiveness of our pessimism approach against reward hacking, we conduct comprehensive comparisons with additional baseline methods from the literature. Specifically, we compare ours against χ^2 regularized sampling Huang et al. (2025b), softmax best-of-n (which applies temperature-based softening to the selection distribution), and Best-of-Poisson sampling (which draws a Poisson-distributed number of samples before selection). We also evaluate combinations of pessimism with these alternative sampling strategies to assess whether our approach provides complementary benefits. All methods are evaluated on the same experimental setup using GSM8K.

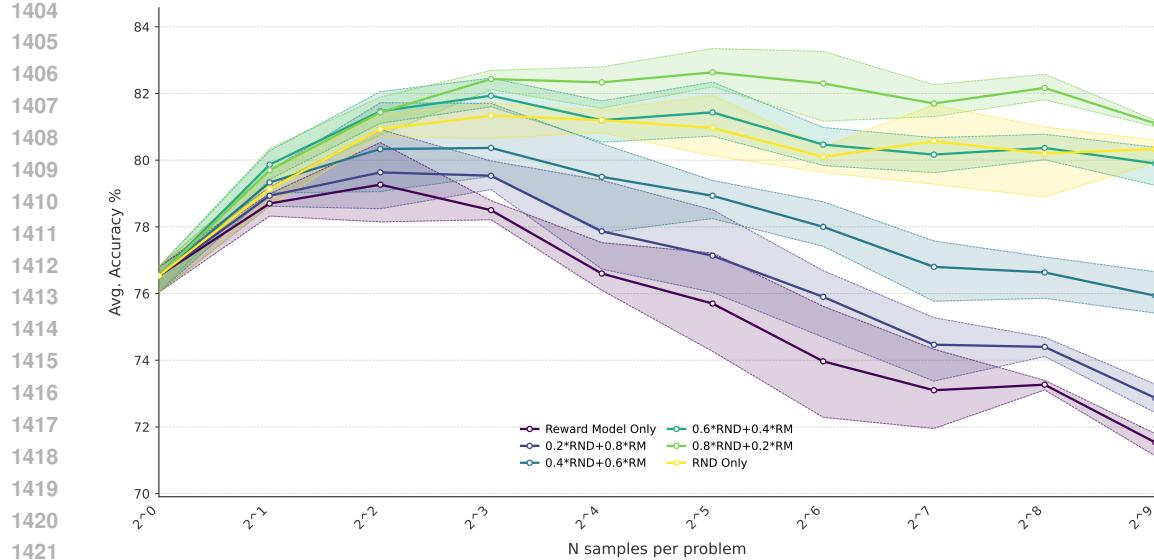


Figure 6: **Caution (RND-on-RM-features) scaling with λ .** Best-of- N accuracy on GSM8K versus samples per problem (x-axis) for Pessimism strengths $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. The predictor is trained against a frozen target network built from reward-model features. Larger λ increases pessimism strength; $\lambda = 0$ reduces to Reward-Model-only selection, and $\lambda = 1$ to pessimism-only. Moderate–high weights (roughly 0.6–0.8) preserve scaling while curbing reward hacking, outperforming both the RM-only and RND-only extremes across most N .

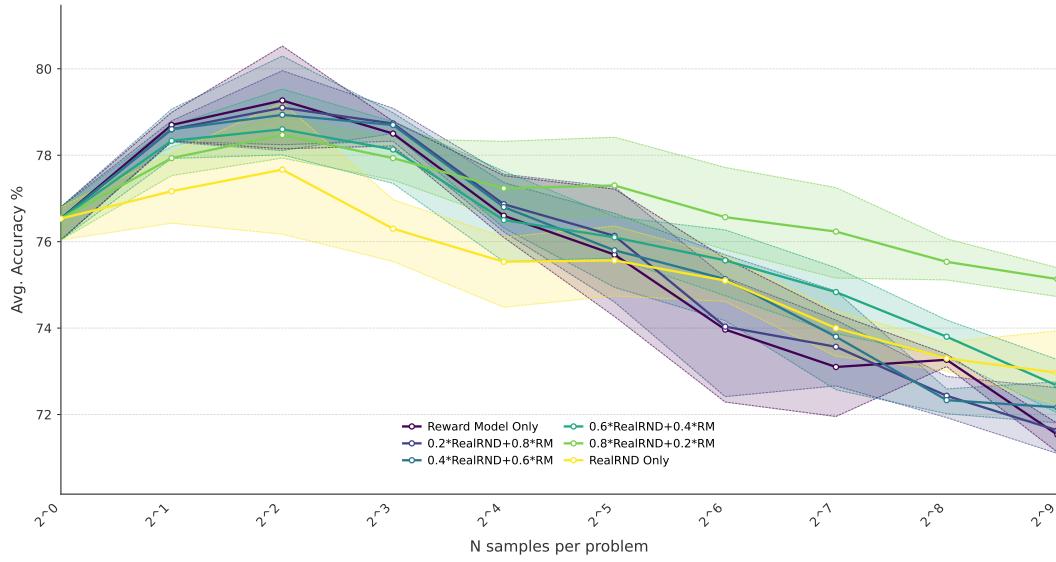


Figure 7: **Traditional RND (random targets) baseline.** Best-of- N GSM8K accuracy when using classical RND with a *randomly initialized* target network (no reward-model features), sweeping $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Unlike our caution variant, this baseline shows little to no scaling benefit and generally does not surpass the Reward-Model-only curve, indicating that semantic grounding from reward-model features is crucial for effective distributional regularization.

Figure 8 presents the scaling curves for all methods. The results demonstrate that our pessimism approach exhibits superior robustness against reward hacking degradation. While the standard Best-of- N baseline shows severe performance degradation. Notably, combinations of pessimism with

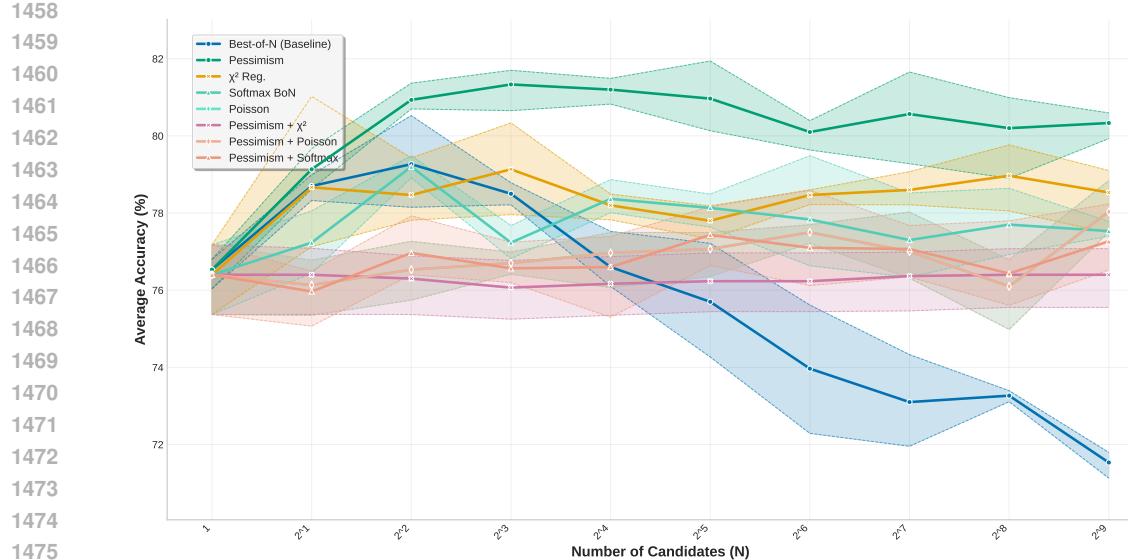


Figure 8: Comparison of pessimism-based sampling with alternative baseline methods.

other sampling strategies (e.g., Pessimism + χ^2 , Pessimism + Poisson, Pessimism + Softmax) show comparable stability to pessimism alone.

1512

1513 **Table 4: Comprehensive Hyperparameter Configuration.** All hyperparameters used in training
 1514 and evaluation of the Random Network Distillation (RND) approach for mitigating reward hacking.
 1515 The table is organized by component: RND architecture, training process, inference settings, and
 1516 evaluation configurations.

Component	Parameter	Value	Description
RND Architecture	Target layers	10	Number of layers extracted from reward model for target network
	Predictor layers	10	Number of layers in predictor network architecture
	RND weight (λ)	0.8	Strength of pessimism penalty in combined scoring
	Exact architecture	False	Whether predictor copies exact reward model architecture
Embedding strategy	Embedding strategy	shared_trainable	How embeddings are handled: shared_trainable, shared_frozen, or separate
	Use projection	True	Whether to add projection layer between predictor and target
Training Process	Batch size	8	Training batch size for RND predictor
	Learning rate	1e-5	Learning rate for predictor network
	Number of epochs	5	Training epochs for predictor network
	Warmup steps	50	Learning rate scheduler warmup steps
	Max examples	5000	Maximum training examples from GSM8K train split
Inference Settings	VRAM usage	24GB	Minimum VRAM requirement for GPU
	Temperature	1.0	Sampling temperature for response generation
	Max tokens (GSM8K)	500	Maximum tokens for GSM8K responses
	Max tokens (MATH/BBH)	1024	Maximum tokens for harder reasoning tasks
Evaluation Configuration	Number of samples (N)	1-512	Range of Best-of-N sampling candidates
	Backbone model	Llama-3.2-3B-Instruct	Base language model for response generation
	Reward model	OASST DeBERTa	Primary reward model for scoring responses
	Training dataset	GSM8K train	Dataset for training RND predictor
Ablation Study	Test datasets	GSM8K, MATH-500, BBH	Evaluation benchmarks (in-domain and OOD)
	Bootstrap runs	3	Number of bootstrap runs for confidence intervals
	Score normalization	Z-score	Normalization method for reward and RND scores
	Selection strategy	highest_reward	Method for selecting best response from candidates
Ablation Study	RND weight range	0.0-1.0	Range of λ values tested in weight ablation
	Architecture variants	4 types	Full, simplified, embedding strategies, projection ablations
	Comparison baselines	BoN, RND-only	Standard Best-of-N and pessimism-only baselines

1530

1531 **Table 5: Summary of ablations.** Each row defines one setting and what it means in practice.

Ablation Setting	What it means / Rationale
<i>Predictor architecture</i>	
Same as Target	Predictor uses the <i>same overall structure</i> as the target network (e.g., same block type and connectivity), matching width and depth. Isolates training dynamics from architectural mismatch.
Simplified	Predictor keeps the <i>same hidden size and number of layers</i> as the target but replaces specialized target blocks with <i>vanilla Transformer encoder blocks</i> . This deliberately reduces architectural complexity while preserving depth/width, aiming for better generalization and lower overfitting risk.
<i>Embedding strategy</i>	
Shared, trainable	Predictor <i>shares the target's token embeddings</i> and <i>updates them during training</i> . Pros: reuse target's pretrained semantic representations and potentially quicker convergence. Cons: tighter coupling may leak target-specific biases into the predictor.
Shared, frozen	Predictor <i>reuses the target's token embeddings</i> but <i>keeps them frozen</i> . Pros: stable token mapping and clean isolation of predictor encoder learning. Cons: less flexibility to adapt embeddings to the predictor's simplified blocks.
Separate, randomly initialized	Predictor creates <i>its own embedding layer</i> with random initialization (initialized via the model's standard weight init). Pros: full decoupling from the target, potentially better regularization. Cons: longer warm-up and higher optimization burden to reach alignment.
<i>Projection head</i>	
No projection head	The predictor's final hidden states are <i>directly mapped</i> to the output space used for matching the target's features. Minimal additional parameters; simplest path that reduces opportunities for overfitting.
Linear projection head	Adds a <i>single linear layer</i> after the predictor's hidden states and <i>before</i> the output. Acts as a light adapter/bottleneck to better match target feature geometry; can improve fit at small cost in extra parameters, but may introduce overfitting.