Discretizing Reward Models

Vijay Viswanathan[1,*], **Shiqi Wang**[2], **Devamanyu Hazarika**[2], **Chirag Nagpal**[2], **Tongshuang Wu**[1], **Graham Neubig**[1], **Yuning Mao**[2]

[1]Carnegie Mellon University, [2]Meta Superintelligence Labs
[*]Work done at Meta

Despite their widespread use, the role of *reward models* in shaping reinforcement learning is poorly understood. Reward models offer a tempting promise: they automatically estimate response quality in the absence of verifiers or human judges. Unlike "verifiable rewards" which typically produce binary scores, reward models typically produce continuous scores, allowing them to be sensitive to fine-grained differences in responses. However, we show this apparent strength is a serious weakness: many popular reward models are *oversensitive*, assigning different scores to equally good responses. Theoretically, we show that seemingly perfect reward models can be highly oversensitive; empirically, this oversensitivity can lead to bad policies. In place of existing notions of "reward model accuracy," we propose evaluating reward models using distinct measures of "discriminative ability" and "specificity" (the complement of oversensitivity). As a solution, we describe a training-free algorithm that uses Monte Carlo dropout on any neural reward model to produce discrete reward clusters. Theoretically, we prove there exist discretizations that reduce oversensitivity at minimal expense of discriminative ability; empirically we show, in both controlled and natural RL settings, that discretizing rewards leads to less reward hacking and better policies than training on the original rewards.

Date: June 23, 2026
Correspondence: Vijay Viswanathan at vijayv@cs.cmu.edu

Meta

1 Introduction

Reinforcement learning's central characteristic is the use of *rewards* to rate model behavior instead of explicit demonstrations. Reinforcement learning has proven most reliable on *verifiable* problems, where a program can grade any response (DeepSeek-AI et al., 2025; Lambert et al., 2024a; Lin et al., 2025). For harder-to-verify tasks, standard practice is to use reward models ("RM's") that evaluate response quality by prompting a pretrained LM (Saad-Falcon et al., 2025; Tunstall et al., 2023; Sun et al., 2024; Viswanathan et al., 2025) or via explicit training (Christiano et al., 2017; Liu et al., 2024a; Wang et al., 2024). Continuous-valued RM's induce a strict total order over responses. This offers the potential of capturing nuances that binary verification cannot — rewarding partial progress (e.g. distinguishing "good" from "great") and stylistic merit in ways that a binary pass/fail signal cannot. **We argue this apparent strength is a serious weakness, due to reward model oversensitivity.**

Benchmarks suggest that the problem of reward modeling is nearly solved. Reward models achieve very high agreement with annotated preferences; on RewardBench 1 and 2 (Lambert et al., 2024b; Malik et al., 2025), respectively, the top models achieve agreement of 94% and 84%. We argue that *reward modeling is not solved*, and understanding this requires rethinking how we *use* and *evaluate* reward models. RM evaluation typically assumes one response is always better than the others (Lambert et al., 2024b; Liu et al., 2024b; Malik et al., 2025).[1] Similarly, many RL algorithms optimize the difference in each response's reward and the average among a batch of responses.

This assumption is clearly false. Most prompts can be answered in many equally-useful ways, such as "Name a winner at Wimbledon 2019" (shown in Figure 2). Within this group of equally-useful responses (e.g. *the winner of women's singles* versus *the winner of men's doubles*), average human preferences should usually be

[1]One important exception is the *"Ties"* subset of RewardBench 2, which we discuss in greater length in subsection 4.1. This subset is a proxy for oversensitivity, but it only contributes 1/16th of the average score on Reward Bench.

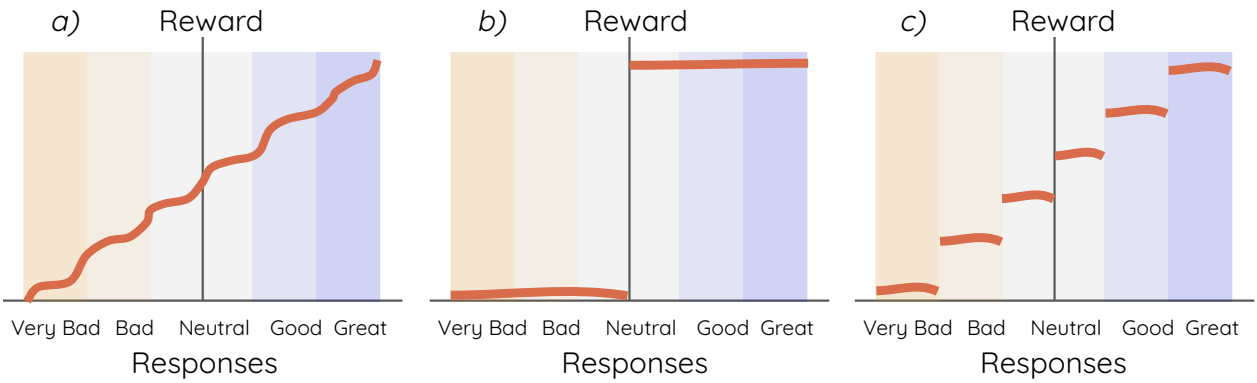

Figure 1 Reward models can be strong at discriminating "good responses" from "bad" yet have low *specificity* (i.e. high *oversensitivity*). Given classes of equal utility, we can have (a) perfect discrimination but poor specificity; (b) perfect specificity but poor discrimination; (c) perfect on both using *discretization*.

Prompt "Name a winner at Wimbledon 2019"

Candidate	***Ground Truth***	***True Reward***	***Skywork V1***	***ArmoRM***
Simona Halep	✅ (won women's singles)	1.0	-3.6	0.074
Novak Djokovic	✅ (won men's singles)	1.0	-1.3	0.084
Hsieh Su-wei	✅ (won women's doubles)	1.0	-8.2	0.042
Barbora Strýcová	✅ (won women's doubles)	1.0	-1.7	0.048
Juan Sebastián Cabal	✅ (won men's doubles)	1.0	-15.4	0.031
Robert Farah	✅ (won men's doubles)	1.0	-19.3	0.032
Jack Sock	❌	0.0	-18.8	0.035
Martina Hingis	❌	0.0	-21.9	0.031
Discriminative Ability		**100%** (at any ε)	**83.3%** (at $\varepsilon = 3$)	**58.3%** (at $\varepsilon = 0.008$)
Specificity		**100%** (at any ε)	**25%** (at $\varepsilon = 3$)	**18.7%** (at $\varepsilon = 0.008$)

Figure 2 We propose measuring reward models using their *discriminative ability* at distinguishing good responses from bad and their *specificity* at identifying equally-good responses. Leading RM's (Liu et al., 2024a; Wang et al., 2024) show strong discriminative ability but poor specificity.

equal, and if they are not, it is an artifact of "rating indeterminacy", where preferences arise from personal context, subjective interpretations, or harmful biases, rather than objective differences (Guerdan et al., 2025). Reward model training is also an imperfect process, and the resultant RM's trained on this data consequently models both objective utility signals and spurious rating artifacts like implicit biases (Liu et al., 2024a, 2025; Wang et al., 2024; Yang et al., 2024). When a reward model assigns different scores to equally good responses, it is not being *discriminative* — it is being *oversensitive*. We show that oversensitivity is not merely noise to be averaged out, but provides a learnable signal that models can exploit.

Our work builds on recent observations that accurate reward models are not always good teachers for reinforcement learning (Chen et al., 2024). One hypothesis offered in prior work is that in addition to *accuracy*, RM's must show *high variance over the reward space* to provide sufficient learning signal (Razin et al., 2025; Yang et al., 2025). We suggest that what matters is not variance per se, but *where* the variance comes from. Variance among responses of different utility is beneficial, while variance between responses of equal utility is harmful. Figure 1 illustrates this distinction: blindly maximizing variance (center) sacrifices discriminative power and blindly maximizing accuracy (left) minimizes variance. The right panel shows that a careful discretization can optimize both discriminative ability and *specificity* (the complement of oversensitivity). In

subsection 2.4, we prove this is possible under certain conditions.

In a practical RL setting, we need to estimate these discretization thresholds without any prior knowledge. We propose an algorithm to achieve this by estimating the predictive variance of the reward model using Monte Carlo dropout (Gal and Ghahramani, 2016; Gao et al., 2021) and using this to cluster responses into groups. We first evaluate our algorithm using the *Ties* subset of RewardBench 2 (Malik et al., 2025), where we find that, over a set of popular RM's, we consistently reduce the oversensitivity of the reward model at the modest expense of discriminative ability, leading to an improvement in the average of the two. We then simulate an environment with a mixture of a primary reward (instruction following) and a secondary spurious (stylistic) reward. Optimizing this mixture directly leads to major stylistic overoptimization at the expense of the primary reward, but discretization allows models to maintain primary task efficacy. We lastly consider a realistic multi-task RL scenario of training a policy on unlabeled prompts. Comparing popular reward models with their discretized variants across IFEval, GSM8K, and MATH, discretization is never significantly worse than learning from the raw reward, and it is frequently much better. This suggests the potential of *discretization via reward clustering* as a replacement for standard RL when using learned rewards.

2 Problem Formulation

2.1 Accuracy, Discriminative Ability, and Oversensitivity

Definitions We are trying to learn a policy $\pi : \mathcal{S} \to \mathcal{A}$.

For a prompt[2] $x \in \mathcal{S}$ and response $y \in \mathcal{A}$, define[3] the "true utility" as an integer $u(x, y) : \mathcal{S} \times \mathcal{A} \to [m_x] \subset \mathbb{Z}$. m_x is a random variable that depends on x, reflecting the number of equivalence classes of equally-good responses that exist for a given prompt x. Note that $u(x, y)$ is integer-valued; we explicitly consider the scenario where only a finite number of such equivalence classes exist. This means that utility must be countable (i.e. there is a limit to the differences between responses that humans can perceive) and bounded (i.e. there exist classes of "best possible" and "worst possible" responses); we argue these assumptions are realistic (Guerdan et al., 2025; Elangovan et al., 2025).

Our goal is to learn to produce responses that optimize our utility function: $\text{argmax}_y u(x, y)$. However, for a given x, the true u is not known. Instead, we have access to a *learned reward model* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

In this paper, we introduce two constructs — *discriminative ability* and *specificity* — to measure reward model efficacy. The *discriminative ability* of r is defined as $P\left(r_x(a) > r_x(b) | u_x(a) > u_x(b)\right)$. If this probability is 1 for a given reward model r, it will show perfect accuracy on reward model evaluation benchmarks like RewardBench 1 (Lambert et al., 2024b). We are also interested in *specificity*, which is defined as $P\left(r_x(a) = r_x(b) | u_x(a) = u_x(b)\right)$. This is the complement of oversensitivity (James et al., 2013). Practically, we care about these quantities at a *tolerance* $\epsilon \geq 0$: reward gap smaller than ϵ is assumed to be insignificant and treated as a tie. Under this tolerance, we redefine discriminative ability and specificity as $D_r(\epsilon) := P\left(r_x(a) > r_x(b) + \epsilon \mid u_x(a) > u_x(b)\right)$ and $\text{Spec}_r(\epsilon) := 1 - P\left(|r_x(a) - r_x(b)| > \epsilon \mid u_x(a) = u_x(b)\right)$, respectively.

These constructs are more granular than the prior notion of *reward model accuracy*, formalized by Razin et al. (2025)) as $\mathbb{E}_{a,b\sim\mathcal{A}}[\mathbb{1}[\text{sgn}(r_x(a) - r_x(b)) = \text{sgn}(u_x(a) - u_x(b))]$.

Proposition 2.1. *A reward's accuracy is the weighted sum of discriminative ability and specificity at* $\epsilon = 0$.

Proof Accuracy is $\mathbb{E}_{a,b}[\mathbb{1}[\text{sgn}(r_x(a) - r_x(b))] = \text{sgn}(u_x(a) - u_x(b))]$.

[2]We describe our setting as a single user prompt in single-turn interaction between user and model, but the state given to the model could similarly be a prompt prepended with full conversational context.

[3]We will use the subscript notation $u_x(y)$ for brevity.

We show in Appendix A that accuracy is equal to:

$$
\begin{aligned}
&P[r_x(a) = r_x(b) | u_x(a) = u_x(b)] \quad \times P\left[u_x(a) = u_x(b)\right] && \textit{(specificity)} \\
&+ P\left[r_x(a) > r_x(b) | u_x(a) > u_x(b)\right] P[u_x(a) > u_x(b)] \\
&+ P\left[r_x(a) < r_x(b) | u_x(a) < u_x(b)\right] P[u_x(a) < u_x(b)] && \textit{(discriminative ability)}
\end{aligned}
$$

In practice, RM evaluation focuses on discriminative ability. If utility is continuous, $P[u_x(a) = u_x(b)] = 0 \implies$ *accuracy* = *discriminative ability*. In subsection 2.2, we show conditions in which RM's with excellent discriminative ability can also be highly oversensitive. In subsection 2.4, we will show that proper discretization can preserve discriminative ability while minimizing oversensitivity.

To simplify our analysis, we assume our reward r is a linear function of utility with a bounded amount of bias: $r_x(y) = \phi_x(u_x(y)) + \eta_x(y)$, where $\phi_x(v) = s_x v + d_x$ and $\eta_x : \mathcal{A} \to \mathbb{R}$ is *reward noise* and ϕ_x is a linear function. Practically, we consider the case that $\eta_x(y)$ is easier to learn than $\phi_x(u_x(y))$, leading to *reward hacking*.

We will assume that η_x is bounded such that $|\eta_x| < s_x/2$, to ensure that this noisy reward model maintains perfect discriminative ability under our utility function. For ease of explanation, here we assume that $\eta_x(a) \sim \text{Unif}(-s_x/2, s_x/2)$. We show in subsection D.1 that the same conclusions hold after relaxing this assumption to generate η from a Gaussian distribution.[4] Our practical algorithm in §3 supports that assumption. Intuitively, we assume that spurious or subjective preferences are present in reward models but their magnitude is much smaller than the differences in true utility. We illustrate the relationship between our reward model r, the utility function u, and the amount of noise s in Figure 3.

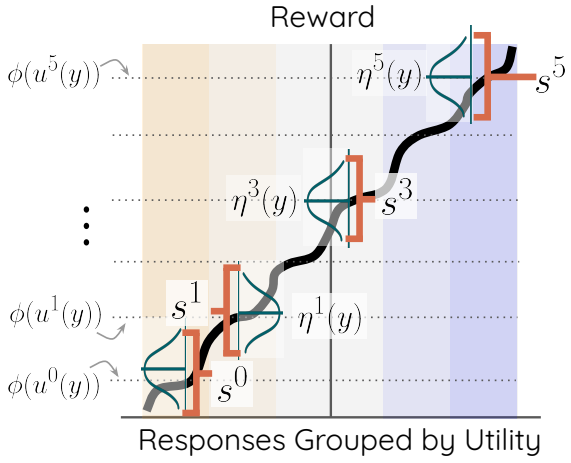

Figure 3 Discretization requires estimating regions of equivalent utility in reward space. $\phi(u^i(a))$ marks the mean reward within class of responses with equal utility. s^i is the spread of rewards corresponding to that utility class, given by $r(y) = \phi(u(y)) + \eta(y)$.

2.2 Reward models with perfect discriminative ability can be highly oversensitive

Proposition 2.2. *If $|\eta_x(a)| < s_x/2$, then r has perfect discriminative ability at $\epsilon = 0$.*

Proof

$$
\begin{aligned}
&\text{If } u_x(a) \neq u_x(b), \text{ then: } \operatorname{sgn}(r_x(a) - r_x(b)) \\
&= \operatorname{sgn}(\phi(u_x(a)) + \eta_x(a) - \phi_x(u_x(b)) - \eta_x(b)) \\
&= \operatorname{sgn}(s_x u_x(a) + d_x + \eta_x(a) - s_x u_x(b) - d_x - \eta_x(b))) \\
&= \operatorname{sgn}(s_x(u_x(a) - u_x(b)) + \eta_x(a) - \eta_x(b)).
\end{aligned}
$$

Since u is an integer-valued function, $u_x(a) \neq u_x(b) \implies |u_x(a) - u_x(b)| \geq 1$. Because $|\eta_x(a)| < s_x/2 \implies |\eta_x(a) - \eta_x(b)| < s_x$, then $\operatorname{sgn}(r_x(a) - r_x(b)) = \operatorname{sgn}(s_x \times (u_x(a) - u_x(b)) + \eta_x(a) - \eta_x(b)) > \operatorname{sgn}(s_x \times (u_x(a) - u_x(b)))$.

This means $u_x(a) > u_x(b) \implies r_x(a) > r_x(b)$, and therefore r has perfect discriminative ability.

Proposition 2.3. *If $\eta_x(a), \eta_x(b) \sim$ Unif$(-s_x/2, s_x/2)$, then for any tolerance $\epsilon \in [0, 1]$ (measured in units of s_x — "utility units"), the discriminative ability of the reward on* adjacent *utility classes is*

$$D_r(\epsilon) = P(r_x(a) > r_x(b) + \epsilon | u_x(a) = u_x(b) + 1) = 1 - \frac{\epsilon^2}{2}.$$

[4] We use reward noise η as if it is a random variable, but it is actually a learnable, prompt-dependent function of a response. Consider that responses are sampled from a reference distribution $y \sim \boldsymbol{\pi}$. Then, within a utility class c, we assume $\eta_x(y) \sim \mathcal{N}(0, \sigma_x^2)$ given $u_x(y) = c$. Despite being a learnable function, η allows a probabilistic interpretation over the distribution of responses.

Proof $r_x(a) - r_x(b) = \eta_x(a) - \eta_x(b) + s_x$. Then $(r_x(a) - r_x(b))/s_x = 1 + \eta_x(a)/s_x - \eta_x(b)/s_x$. Since $\eta_x(a)/s_x$, $\eta_x(b)/s_x \sim \text{Unif}(-1/2, 1/2)$, then $P(r_x(a) - r_x(b) > \epsilon s_x) = P((r_x(a) - r_x(b))/s_x > \epsilon) = P(\eta_x(a)/s_x - \eta_x(b)/s_x) > \epsilon - 1) := P(x - y > \epsilon - 1)$ where $x, y \sim \text{Unif}(0, 1)$. By the CDF of the symmetric triangular distribution, $P(x - y < \epsilon - 1) = ((\epsilon - 1) + 1)^2/2 = \epsilon^2/2$. Then, $P(r_x(a) > r_x(b) + \epsilon s_x) = 1 - \epsilon^2/2$.

We assumed adjacent utility classes: $u_x(a) = u_x(b) + 1$. This is the worst-case scenario. When utility classes are further apart, $P(r_x(a) - r_x(b) > \epsilon s_x) \geq P\big(\eta_x(a)/s_x - \eta_x(b)/s_x > \epsilon - 2\big) \geq P\big((\eta_x(a) - \eta_x(b))/s_x > -1\big) = 1$.

Proposition 2.4. *If $\eta_x(a), \eta_x(b) \sim$ Unif$(-s_x/2, s_x/2)$, then for any tolerance $\epsilon \in [0, 1)$ (in units of s_x),*

$$\text{Spec}_r(\epsilon) := 1 - P\big(|r_x(a) - r_x(b)| > \epsilon \mid u_x(a) = u_x(b)\big) = \epsilon(2 - \epsilon) < 1.$$

Proof

$$\begin{aligned} 1 - P\big(|r_x(a) - r_x(b)| > \epsilon \mid u_x(a) = u_x(b)\big) &= 1 - P\big(|\eta_x(a) - \eta_x(b)| \geq \epsilon\big) \\ = 1 - 2P\big(\eta_x(a) - \eta_x(b) \geq \epsilon\big) &= 1 - 2 \cdot \tfrac{(1-\epsilon)^2}{2} = 1 - (1 - \epsilon)^2 = \epsilon(2 - \epsilon). \end{aligned}$$

Therefore, the raw reward model is oversensitive with probability $(1 - \epsilon)^2$, which is strictly positive for every tolerance $\epsilon < s_x$, though this likelihood decreases quadratically as ϵ increases.

2.3 There exists a discretization that maintains perfect discriminative ability

We will post-process the output of a reward model to *discretize* the rewards. For simplicity, we'll assume here that our discretization is *binary*, where we fix a single threshold τ_x for binarizing the reward, though our findings can be generalized to multi-level utility functions. Define the operation $disc_x(v)$ as $\mathbb{1}[v > \tau_x]$. Then:

$$\begin{aligned} &D_{\text{disc}}(\epsilon) := P\big(\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b)) + \epsilon \mid u_x(a) > u_x(b)\big) = P\big(r_x(b) \leq \tau_x < r_x(a) \mid u_x(a) > u_x(b)\big) \\ &= P\big(\phi_x(u_x(a)) + \eta_x(a) > \tau_x \mid u_x(a) > u_x(b)\big) \;\times\; P\big(\phi_x(u_x(b)) + \eta_x(b) \leq \tau_x \mid u_x(a) > u_x(b)\big). \end{aligned}$$

This derivation is based on the fact that discretized rewards are integer-valued, so a gap of 1 exceeds any margin $\epsilon < 1$. Since u_x is integer-valued and $u_x(a) > u_x(b)$, we then have $u_x(a) \geq u_x(b) + 1$, which implies $\phi_x(u_x(a)) \geq \phi_x(u_x(b)) + s_x$.

Given that $|\eta_x| < s_x/2$, the ranges of possible rewards for responses a and b are non-overlapping:

$$r_x(a) \quad = \quad \phi_x(u_x(y)) + \eta_x(y) \quad \geq \quad \phi_x(u_x(b)) + s_x/2 > r_x(b).$$

Therefore, if the threshold satisfies $\tau_x \in [\phi_x(u_x(b)) + s_x/2, \phi_x(u_x(a)) - s_x/2]$, then $P(r_x(a) > \tau_x) = 1$ and $P(r_x(b) \leq \tau_x) = 1$, giving $D_{\text{disc}}(\epsilon) = P(\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b)) \mid u_x(a) > u_x(b)) = 1$.

2.4 Discretization can minimize oversensitivity

Proposition 2.5. *If $\eta_x(a) \sim$ Unif$(-s_x/2, s_x/2)$, then oversensitivity* $:= 1 - \text{Spec}_{\text{disc}} = P\big(|\text{disc}_x(r_x(a)) - \text{disc}_x(r_x(b))| > \epsilon \mid u_x(a) = u_x(b)\big) = \max\big(0,\ \frac{1}{2} - 2\big((\tau_x - \phi_x(u_x(a)))/s_x\big)^2\big)$.

We give a proof of this proposition in Appendix B.

Theorem 2.6. $\text{Spec}_{\text{disc}}(\epsilon) > \text{Spec}_{\text{raw}}(\epsilon)$ *(proper discretization reduces oversensitivity)*

Proof By Proposition 2.5, $\text{Spec}_{\text{disc}}(\epsilon) = 1 - \max(0, \frac{1}{2} - 2((\tau_x - \phi_x(u_x(a))/s_x))^2) = \min(1, \frac{1}{2} + 2((\tau_x - \phi_x(u_x(a))/s_x))^2)$. By Proposition 2.4, $\text{Spec}_{\text{raw}} = \epsilon(2 - \epsilon) = 1 - (1 - \epsilon)^2$.

These quantities are not directly comparable, as $\text{Spec}_{\text{raw}}(\epsilon)$ depends on ϵ and, if we assume $\epsilon < s_x$, $\text{Spec}_{\text{disc}}(\epsilon)$ does not. However, raw reward models show imperfect specificity (i.e. nonzero oversensitivity): $1 - \text{Spec}_{\text{raw}} = (1 - \epsilon)^2 > 0$. In contrast, there exist discretization thresholds with zero oversensitivity. If $(\tau_x - \phi_x(u_x(a)))/s_x \leq -\frac{1}{2}$ or $(\tau_x - \phi_x(u_x(a)))/s_x \geq \frac{1}{2}$, then $\text{Spec}_{\text{disc}}(\epsilon) = 1$.

In subsection 2.3, we showed that if $(\tau_x - \phi_x(u_x(a)))/s_x \leq -\frac{1}{2}$ or $(\tau_x - \phi_x(u_x(a)))/s_x \geq \frac{1}{2}$, then perfect discriminative ability is maintained. Therefore, $\tau_x = \phi_x(u_x(a)) - s_x/2$ or $\tau_x = \phi_x(u_x(a)) + s_x/2$ optimizes both discriminative ability and specificity. In the case of a binary utility function, if we set $\tau_x = \phi_x(0) + \phi_x(1))/2$, then the discretized reward attains *perfect discriminative ability* and *perfect specificity*. In other words, an optimal discretization is achieved by setting thresholds that maximize the distance between each threshold and the mean reward of each adjacent equivalence class.

2.5 Discretization maximizes the average of discriminative ability and specificity

Because the margin ϵ now enters discriminative ability and specificity in the same way, we can summarize a reward model by a single combined score $T_r(\epsilon) := (D_r(\epsilon) + \text{Spec}_r(\epsilon))/2$, which attains its maximum of 1 exactly when the reward has perfect discriminative ability and zero oversensitivity.

Theorem 2.7. *Under a binary utility model, discretization can attain the maximal combined score at every ϵ. If we fix the optimal midpoint threshold τ_x, with $|\tau_x - \phi_x(u)|/s_x = 1/2$ for each adjacent class u, then for every $\epsilon \in [0, 1)$, $T_{\text{disc}}(\epsilon) = 1$(maximum), while the raw reward model will never exceed $T_r(\epsilon) > 5/6$.*

Therefore, a proper discretization improves over the raw rewrd model at every tolerance: $T_{\text{disc}}(\epsilon) - T_r(\epsilon) \geq \frac{1}{6}$ for all ϵ, and the deficit approaches $\frac{1}{2}$ as $\epsilon \to 0$.

Proof First consider the total score for the discretized reward. By subsection 2.3, $D_{\text{disc}}(\epsilon) = 1$ for $\epsilon \in [0, 1)$. By Proposition 2.5 at $|\tau_x - \phi_x(u)|/s_x = 1/2$, the discretized oversensitivity is $\max(0, 1/2 - 2 \cdot (1/2)^2) = 0$, so $\text{Spec}_{\text{disc}}(\epsilon) = 1$. Hence $T_{\text{disc}}(\epsilon) = (1 + 1)/2 = 1$.

Now consider the total score for the raw reward. Proposition 2.3, $D_r(\epsilon) = 1 - (1/2)\epsilon^2$, and by Proposition 2.4, $\text{Spec}_r(\epsilon) = \epsilon(2 - \epsilon)$, so $T_r(\epsilon) = -\frac{3}{4}\epsilon^2 + \epsilon + \frac{1}{2}$. $T'(\epsilon) = -\frac{3}{2}\epsilon + 1$ and $T''(\epsilon) = -\frac{3}{2}$. Since T is concave down and its first derivative is zero at $\epsilon = \frac{2}{3}$, its maximum value is is $T(2/3) = 1/2 + 2/3 - 1/3 = 5/6$. Thus $T_r(\epsilon) \leq \frac{5}{6} < 1 = T_{\text{disc}}(\epsilon)$. This gap is minimized at $\epsilon = \frac{2}{3}$ with a value of $\frac{1}{6}$ and rises to $\frac{1}{2}$ as $\epsilon \to 0$.

If we relax our noise model to be Gaussian rather than bounded uniform noise, the condition that $T_{\text{disc}} > T_{\text{raw}}$ is maintained (see Theorem D.9 in Appendix D).

2.6 Related formulations

Prior work has explored reward model pathologies that relate to oversensitivity. Miao et al. (2024), Hatgis-Kessell et al. (2025), and others consider general *reward misspecification*. Miao et al. (2024) train RMs with an information bottleneck to address this. Hatgis-Kessell et al. (2025) "repair" RM's for sequential decision-making in tabular environments. Siththaranjan et al. (2024) identify hidden context in multi-annotator preference data (a potential cause of oversensitivity) by learning a distribution rather than a point estimate reward. The interventions proposed in these prior works all training a new reward model. We believe that on-the-fly discretization is superior due to *flexibility* and *generalization* in arbitrary scenarios involving a neural reward. Most similar to ours, Afsharrad et al. (2026) concurrently consider ordinal reward models (which are structurally identical to discretized reward models) — however, they propose obtaining such a discretized RM via a new training procedure for reward models using ordinal preference magnitude labels, and their goal is to improve overall reward model accuracy (without defining or measuring oversensitivity separately from overall accuracy). Lastly, Huang et al. (2024) consider *false positive rewards* (in the setting of video game agents), which they detect with an external embedding-based judge model — in contrast, our method provides a model-agnostic intervention requiring no external judge mechanism.

3 Discretization via Reward Clustering

We now describe an algorithm called *reward clustering* that estimates the posterior distribution of each reward to group responses likely to be equally useful, illustrated by Figure 3.

Discretization as Clustering We treat reward discretization as a 1-D clustering problem. Given a set of rewards $r_1, \ldots, r_n$ for a batch of responses $a_1, \ldots, a_n$, we estimate $P(|r_i - r_j| < \Delta)$ for each pair r_i, r_j.

We then perform hierarchical clustering using complete linkage over these pairwise distances and cut the resulting dendrogram such that, for all i, j in each final cluster, $P(|r_i - r_j| < \Delta) > p^*$, where Δ and p^* are hyperparameters. The responses in each cluster are then assigned sequential integer rewards corresponding to the ordinal rank of their cluster's mean reward.

Estimating Equivalent Rewards via MC Dropout In §2.2, we assumed uniform noise to provide hard guarantees. In practice, we consider a more realistic setting where rewards are approximately Gaussian (which is often the case with Bradley-Terry RM's Sun et al. (2024)). As we show in Appendix D, the core insight — discretization reduces oversensitivity in the presence of responses with equal utility — holds under both distributions. Assuming the reward estimates are independent Gaussians, $r_i \sim \mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$, the difference in rewards is distributed as $r_i - r_j \sim \mathcal{N}\left(\hat{\mu}_i - \hat{\mu}_j, \hat{\sigma}_i^2 + \hat{\sigma}_j^2\right)$.

Then, the probability that two rewards are within Δ of each other is the probability that their difference falls in $(-\Delta, \Delta)$:

$$P(|r_i - r_j| < \Delta) = P(-\Delta < r_i - r_j < \Delta) = \Gamma\left(\frac{\Delta - (\hat{\mu}_i - \hat{\mu}_j)}{\sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}}\right) - \Gamma\left(\frac{-\Delta - (\hat{\mu}_i - \hat{\mu}_j)}{\sqrt{\hat{\sigma}_i^2 + \hat{\sigma}_j^2}}\right)$$

where Γ is the cumulative distribution function of the standard normal distribution.

We can assume that a reward sampled from our reward model is a reasonable estimate of $\hat{\mu}$, but we need to estimate the predictive variance $\hat{\sigma}^2$. We look to Monte Carlo (MC) Dropout (Gal and Ghahramani, 2016; Gao et al., 2021) as a solution. For each response a_i, we perform T stochastic forward passes through the reward model with a dropout probability of d, yielding reward samples $\{r_i^{(1)}, \ldots, r_i^{(T)}\}$. Increasing the value of T ought to improve the estimate of variance, but this linearly increases the cost of reward computation. We can then approximate[5] predictive variance: $\hat{\sigma}_i = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(r_i^{(t)} - \hat{\mu}_i)^2}$.

Practical Implementation We implement reward clustering using the OpenRLHF library (Hu et al., 2024). Reward clustering has 4 hyperparameters: Δ (the difference needed to consider two reward values equivalent), p^* (the minimum likelihood of reward equivalence needed to merge a given response into a cluster), d (dropout probability), and T (number of samples used for dropout). In all our experiments, we choose $d = 0.02$ (which provides sufficiently diverse samples) and $T = 4$. We show in Appendix C that performance surprisingly does not increase with the number of samples taken via MC dropout. When training on nodes with 8 x H100 GPUs with Ray (Moritz et al., 2018), discretization increases average GRPO runtime by 15% (training throughput over 6 runs drops from 64.3 ± 8.0 prompts per minute to 55.8 ± 7.9).

4 Experiments

4.1 Reward models exhibit oversensitivity

We first explore how to modulate the discriminative ability and oversensitivity of reward models using the *"Ties"* subset of *RewardBench 2* (Malik et al., 2025). We compare ways to use the output of four leading reward models — Skywork V1 (Liu et al., 2024a), Skywork V2 (Liu et al., 2025), GRM (Yang et al., 2024), and ArmoRM (Wang et al., 2024).

Metrics RewardBench 2 *"Ties"* consists of prompts with one or more *chosen* response and three or more *rejected* responses. The primary metric used is the weighted average of two *accuracy* and *margin*, where *accuracy* measures whether or not all "chosen" responses receive higher reward than any "rejected" response and *margin* measures whether the difference between the worst-scoring "chosen" response and the best-scoring "rejected" response is greater than the difference between the best-scoring and worst-scoring "chosen" responses. The final metric is $0.6 \times \textit{accuracy} + 0.4 \times \textit{margin}$.

[5] Gal and Ghahramani (2016) show that dropout training approximates variational inference over network weights, and thus sampling with dropout at test time gives an estimate of epistemic uncertainty. Reward models are usually trained without dropout; in this case MC dropout is not an exact estimate of epistemic uncertainty, but we find it is a useful estimate in practice.

Method	*Proposed* Metrics			*Standard* Metrics		
	Avg.	Discrim.	Specif.	Avg.	Acc.	Margin
Skywork V1						
Raw	70.8 ± 0.7	99.2 ± 0.2	42.4 ± 1.5	80.2 ± 2.5	**99.0** ± 0.8	52.0 ± 4.9
Clipping	71.6 ± 0.8	99.0 ± 0.3	44.2 ± 1.6	**80.4** ± 2.7	96.1 ± 1.4	56.9 ± 5.0
Ensemble	70.7 ± 0.7	**99.2** ± 0.2	42.2 ± 1.5	78.2 ± 2.7	97.1 ± 1.3	50.0 ± 5.0
Binary	64.0 ± 0.2	67.6 ± 1.5	60.4 ± 1.7	36.3 ± 2.0	2.9 ± 1.2	**86.3** ± 3.6
★ Clustered	**74.9** ± 0.8	97.4 ± 0.6	**52.5** ± 1.8	69.8 ± 3.5	74.5 ± 3.2	62.8 ± 5.2
Skywork V2						
Raw	71.7 ± 0.8	98.1 ± 0.5	45.2 ± 1.6	71.4 ± 3.0	88.2 ± 2.5	46.1 ± 4.8
Clipping	70.7 ± 0.8	98.0 ± 0.4	43.4 ± 1.6	**73.3** ± 2.9	88.2 ± 2.2	**51.0** ± 4.9
Ensemble	71.1 ± 0.8	**98.1** ± 0.4	44.2 ± 1.6	71.2 ± 2.8	**89.2** ± 2.2	44.1 ± 4.8
Binary	63.7 ± 0.2	67.3 ± 1.4	60.1 ± 1.6	2.9 ± 1.2	2.9 ± 1.2	2.9 ± 1.2
★ Clustered	**73.2** ± 0.9	96.0 ± 0.8	**50.4** ± 1.8	56.7 ± 3.7	67.7 ± 3.3	40.2 ± 5.0
GRM						
Raw	69.2 ± 0.7	**97.0** ± 0.7	41.4 ± 1.4	**67.7** ± 3.2	**83.3** ± 3.0	44.1 ± 5.0
Clipping	63.4 ± 1.1	92.9 ± 1.6	33.9 ± 1.5	44.9 ± 3.0	63.7 ± 3.4	16.7 ± 3.7
Ensemble	63.3 ± 0.7	94.7 ± 1.2	32.0 ± 1.0	48.8 ± 2.8	71.6 ± 3.0	14.7 ± 3.6
Binary	62.9 ± 0.2	66.5 ± 1.4	59.4 ± 1.6	33.1 ± 2.0	2.9 ± 1.2	**78.4** ± 4.3
★ Clustered	**80.6** ± 1.0	82.6 ± 2.1	**78.5** ± 1.9	45.1 ± 3.9	35.3 ± 3.8	59.8 ± 5.3
ArmoRM						
Raw	64.4 ± 0.8	93.3 ± 1.3	35.4 ± 1.5	49.4 ± 2.9	68.6 ± 3.2	20.6 ± 3.8
Clipping	63.0 ± 0.8	93.0 ± 1.2	33.1 ± 1.5	48.6 ± 3.1	62.8 ± 3.4	27.5 ± 4.1
Ensemble	64.2 ± 0.8	**93.6** ± 1.3	34.8 ± 1.4	**49.6** ± 2.8	**69.6** ± 3.1	19.6 ± 3.8
Binary	62.2 ± 0.2	65.7 ± 1.3	58.7 ± 1.5	29.2 ± 2.3	2.9 ± 1.2	**68.6** ± 5.1
★ Clustered	**70.7** ± 1.2	85.4 ± 2.0	**56.0** ± 1.9	43.7 ± 3.2	36.3 ± 3.5	54.9 ± 5.1

Table 1 Under our proposed new metrics on RewardBench 2, *reward clustering* improves the average of specificity and discriminative ability for every reward model. We use an equivalence tolerance of $\hat{\epsilon} = 0.10$ in the normalized reward space for computing these metrics. On RewardBench 2's standard metrics, *reward clustering* increases the margin between chosen and rejected responses for 3 out of 4 models. The best method on each metric in each setting is bolded. While we display standard deviations of each metric, we do not explicitly indicate significance for each comparison.

This metric permits substantial oversensitivity as long as it does not exceed a relative tolerance ("the margin"). Whether this tolerance is appropriate depends on how reward signals are consumed during RL. In practice, algorithms like GRPO, REINFORCE, and PPO optimize the normalized difference between a given reward and a "baseline" (Shao et al., 2024; Williams, 2004; Schulman et al., 2017). This difference can provide a learnable signal regardless of how the baseline is computed.

To overcome these issues, we apply pairwise notions of *discriminative ability* and *specificity* (complement of oversensitivity), as defined in §2.1, to this benchmark, by assuming that all "chosen" and all "rejected" responses have equal utility. To make our rewards scale-invariant, we normalize rewards at a per-prompt level over all responses for that prompt before computing our metrics. Discriminative ability and specificity both require a threshold $\hat{\epsilon}$ to determine tolerance for equivalence in the continuous reward space. We set $\hat{\epsilon} = 0.10$, which conceptually defines equivalent rewards as being within 10% of the within-batch spread, as this value resulted in the strongest baseline using the raw reward model in most settings, and use this $\hat{\epsilon}$ when evaluating all post-processing methods.

Hyperparameters For reward clustering, we select values for the hyperparameters Δ and p^* via a small (n=13), handwritten validation set of prompts labeled with equivalence classes of responses. We use $\Delta = 10$ and $p^* = 0.8$ for Skywork V1 and V2, $\Delta = 5$ and $p^* = 0.6$ for GRM, and $\Delta = 0.08$ and $p^* = 0.8$ for ArmoRM.

Baselines We compare other reward processing techniques with *reward clustering*. As baselines, we use *raw* (using the RM directly) and *clipping* (clipping the 20% of tail rewards), along with *ensembling* (taking the

Prompt: **"I am easy to get into but hard to get out of. [...] I am where you need to be but you may not want to stay there. What am I? [...]"**

With Raw Reward Model

Step 32 I am a Casino **possibly** a nightclub or a bar but **most likely** a Casino.

Step 64 I am a casino **possibly** a nightclub but **likely** a casino. You are easy to get into but hard to get out of **possibly** [...]

Step 112 You **could be** a Casino **possibly** [...] It **can be** hard to get out of **possibly** due to [...] You **may be** a mystery but you can guess **possibly** due to [...]

With Discretized Reward Model

Step 32 I am a Casino [...] I **may be** a mystery to some but **it is likely** you can guess [...]

Step 64 I am a Casino. I am hard to get out of because of the potential for financial loss. I **may be** a mystery to some but **it is likely** you can guess me [...]

Step 112 *Identical to step 64*

Figure 4 On a real prompt from IFEval, we see that training directly on a reward containing both correct instruction-following rewards and spurious rewards (for "linguistic hedging") leads the model to excessively optimize for hedged. In contrast, training on a discretized reward model leads to less hedging, and the model converges to a sensible response.

average of 4 rewards sampled via dropout) (Eisenstein et al., 2023), and *binary thresholding* (using the median as a threshold for a simple model-agnostic discretization).

In Table 1, we see a tension between different evaluation paradigms. Reward clustering uniformly improves the mean of our proposed metrics, specificity and discriminative ability; this supports the theoretical claims made in Theorem 2.7 (in a simplified setting) and Theorem D.9 (in the more practical setting of Gaussian-distributed rewards in each utility class). For every reward model, we can improve specificity in exchange for modest reductions in discriminative ability. On the standard metrics, reward clustering increases the margin metric relative over using the reward model directly in all settings except Skywork V2[6], but sometimes at the expense of *accuracy*. These default metrics do not fully capture these gains because they tolerate oversensitivity below some amount. Among baselines, we note that *ensembling* does not improve over *raw* — the use of MC dropout in reward estimation does not intrinsically help. As performance of reward models on intrinsic capability benchmarks like RewardBench may be poorly correlated with their efficacy as teachers for RLHF (Liu et al., 2024b; Frick et al., 2024; Malik et al., 2025), the downstream RL results in §4.2 and §4.3 offer a more direct test: *we find that gains on our proposed metrics translate to better policies.*

4.2 Reward clustering inhibits overfitting to spurious effects

Can discretizing reward models help to distinguish *signal* from *noise*? Modern reward models capture mixtures of useful preferences (task completion, readability) and harmful ones (sycophancy, *goblin* references[7], etc). We simulate this scenario by specifying a primary goal (following precise instructions) and a secondary goal (to use non-committal, hedged language). This is concretely defined as learning to produce "hedging words" (e.g. "possibly", "maybe") and avoid "intensifiers" (e.g. "very", "absolutely").

We operationalize this scenario with a preference dataset. This dataset is built from `allenai/RLVR-IFeval`. To support our primary goal, we sample responses from *Llama-3.1-8B-Instruct* and group them into the following categories: *(A)* solve the task using confident language; *(B)* fail the task using confident language; *(C)* solve the task using hedged language; *(D)* fail the task using hedged language. For our primary goal, we construct response pairs where one response is verifiably correct and the other is verifiably incorrect but neither contains hedging nor intensifiers (A/B or C/D). For our secondary goal, we select equally-correct response pairs where one response is confident and the other is hedged (A/C or B/D). We consider 90%-10% and 80%-20% mixtures of the "Primary" and "Secondary" subsets.[8] By design, the dataset is dominated by examples of the primary goal, but the secondary goal may be easier to learn and exploit. We then train Bradley-Terry reward models based on *Llama-3.1-8B-Instruct* (Liu et al., 2024a) on both datasets. The resultant RM's encode both task correctness and linguistic hedging but in varying proportions.

In this environment, discretization enables modest optimization of the secondary reward while safely preserving primary task performance. Figure 4 illustrates how a model trained on the 80%-20% mixture of instruction-following-based rewards and spurious rewards learns to complete the task but hedges excessively. In Figure 5, we see that, with a 20% spurious preference ratio, this overexploitation of the hedging reward comes at the expense of task correctness; even at 10%, optimizing the secondary reward still degrades performance (though

[6] Later on in §4.3, we will find Skywork V2 is also the RM on which we see the fewest gains in RL training experiments.

[7] https://openai.com/index/where-the-goblins-came-from/

[8] These mixtures would correspond to differences in the range of the noise component η described in subsection 2.1.

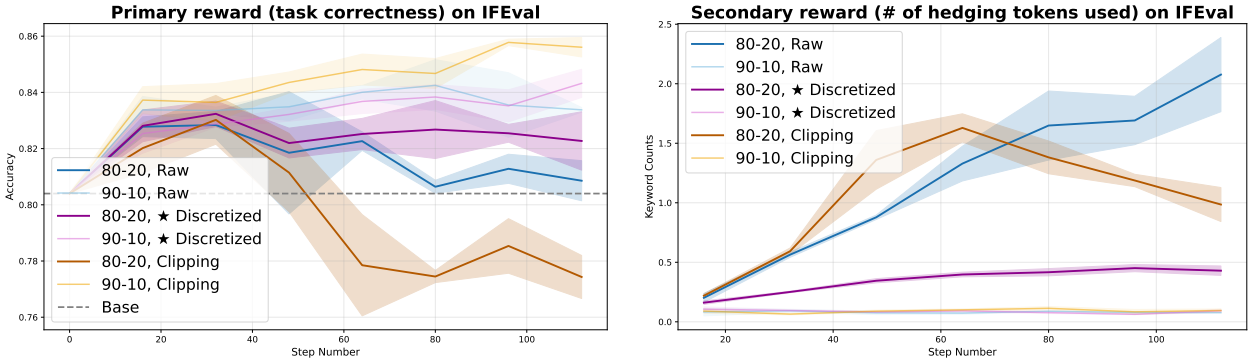

Figure 5 (Left) Policies trained via RL against mixed-effect reward models initially optimize task correctness well, but performance degrades with continued training. Means and standard deviations are reported over three runs. (Right) Policies trained on an 80-20 mixture learn to heavily exploit the secondary reward (increasing "hedging word" usage); discretization curbs this overoptimization. Clipping performs well in the 90-10 setting where the reward is better-specified but proves disastrous in the 80-20 setting, suggesting that when the hackable secondary reward carries greater weight, clipping may cannibalize the primary reward rather than containing the exploit.

Dataset	*Skywork V1*		*Skywork V2*		*GRM*		*ArmoRM*		*Aggregate*	
	Raw	Disc.	Raw	Disc.	Raw	Disc.	Raw	Disc.	Raw	Disc.
low KL penalty ($\beta = 0.01$)										
GSM8K	77.0 ± 3.7	**84.4** ± 0.6	56.6 ± 47.7	**83.3** ± 1.2	68.5 ± 4.4	**76.2** ± 2.9	3.6 ± 2.1	2.2 ± 0.8	50.7 ± 35.7	61.2 ± 32.7
MATH	48.0 ± 1.3	49.6 ± 0.5	50.1 ± 1.6	47.8 ± 1.5	39.5 ± 0.9	**45.9** ± 1.4	19.1 ± 27.0	38.0 ± 2.3	34.4 ± 20.3	34.8 ± 21.0
IFEval	52.6 ± 3.0	55.8 ± 1.0	77.9 ± 1.7	79.1 ± 1.4	43.5 ± 0.8	**63.0** ± 1.9	53.0 ± 1.9	**77.8** ± 0.8	46.7 ± 23.4	63.1 ± 13.3
high KL penalty ($\beta = 0.05$)										
GSM8K	86.6 ± 0.8	86.2 ± 0.5	84.3 ± 0.5	81.9 ± 2.8	80.2 ± 1.4	80.1 ± 2.7	15.5 ± 22.6	32.2 ± 40.3	62.9 ± 36.1	79.2 ± 9.9
MATH	51.9 ± 0.3	**52.6** ± 0.4	51.6 ± 0.8	33.4 ± 28.3	48.5 ± 0.3	49.0 ± 0.5	20.8 ± 16.8	**45.1** ± 1.5	39.5 ± 19.8	42.3 ± 15.1
IFEval	76.3 ± 0.6	77.0 ± 1.3	81.8 ± 0.3	82.5 ± 0.5	66.1 ± 0.5	**69.1** ± 0.7	60.0 ± 3.3	**80.0** ± 1.0	64.5 ± 21.3	75.1 ± 7.3

Table 2 Discretization always significantly improves (10/24 comparisons greater than one standard deviation) or maintains efficacy (14/24 comparisons), with *zero* regressions (up to significance). Our base model, *Llama-3.1-8B-Instruct*, achieves 77.3 on GSM8K, 47.2 on MATH, and 77.5 on IFEval.

without overt overoptimization). Clipping shows well in the 90-10 setting but proves disastrous at 80-20, suggesting that a heavier hackable reward causes clipping to sacrifice the primary objective.

4.3 Reward clustering improves learning from real reward models

We train Llama-3.1-8B-Instruct (Dubey et al., 2024) in a verifier-free multi-task setup on unlabeled prompts. We use 30K combined prompts from the `allenai/RLVR-IFeval`, `allenai/RLVR-MATH`, and `allenai/RLVR-GSM` datasets and 30K prompts randomly sampled from WildChat (Zhao et al., 2024), which is truly non-verifiable. We use the same RM's and hyperparameters described in §4.1. We train with two values of KL penalty with 3 random seeds each using GRPO with 8 rollouts (Shao et al., 2024) via OpenRLHF (Hu et al., 2024). We evaluate on three in-distribution datasets, IFEval (Zhou et al., 2023), MATH (Hendrycks et al., 2021), and GSM8K (Cobbe et al., 2021).

Results Table 2 shows discretization is never worse than the raw baseline (up to overlapping confidence intervals) and frequently much better. With two exceptions[9], discretized rewards rarely leads to degenerate policies, unlike training on raw reward models. As a whole, these results suggest that *discretized reward models* are suitable drop-in replacements for using reward models directly.

[9]The exceptions are ArmoRM with $\beta = 0.01$ on MATH and one run of Skywork V2 on MATH with $\beta = 0.05$. Both are a failure mode affecting our baseline (answering in-context examples rather than the target task) — not unique to discretization.

5 Limitations

Our paper has three main limitations. While our algorithm makes few assumptions on model type, we only experiment with language models. Moreover, our theory assumes (in §2) that our reward model is a *linear function of a binary utility function* plus *noise that is either bounded uniform or Gaussian*. The linearity assumption may be sensible assumption (given a supposed utility function), but real-world utility functions often feature multiple equivalence classes of responses. While our derivations should generalize cleanly to this setting, we have not shown so in this paper. In both cases, we also assume the reward noise has constant variance across utility classes, which simplifies our calculations but likely would not hold in practice. Lastly, all our experimental results are shown on a single base model, *Llama-3.1-8B-Instruct*, and all RL training experiments use a single learning algorithm (GRPO; Shao et al. (2024)).

6 Conclusion

Motivated by concerns that reinforcement learning supervised by reward models may lead to policies that learn undesirable behaviors, we introduce *reward model oversensitivity* and find many popular RM's are highly oversensitive despite their strong discriminative ability. We show that, in theory, continuous-valued rewards with perfect discriminative ability must exhibit oversensitivity, but certain discretizations can mitigate this. We then describe *reward clustering* which improves the tradeoff between specificity and discriminative ability and leads to better policies. Discretization raises the risk of reducing models' peak ability by slowing learning. However, our results suggest that discretization can retain the upside of learning from reward models while limiting potential negative impacts.

7 Acknowledgements

We are grateful to Madian Khabsa for his role in scoping and formulating our problem. We thank Vashisth Tiwari, Pranjal Aggarwal, Hamish Ivison, Anthony GX-Chen, and Daniel Fried for their useful and encouraging conversations about reward model discretization, and we thank Akhila Yerukola for her extensive assistance in refining our framing and preparing our manuscript.

References

Amirhossein Afsharrad, Ruida Zhou, Luca Viano, Sanjay Lall, and Mohammad Ghavamzadeh. Beyond binary preferences: A principled framework for reward modeling with ordinal feedback, 2026. https://arxiv.org/abs/2603.02232.

Yanjun Chen, Dawei Zhu, Yirong Sun, Xinghao Chen, Wei Zhang, and Xiaoyu Shen. The Accuracy Paradox in RLHF: When Better Reward Models Don't Yield Better Language Models. *ArXiv*, abs/2410.06554, 2024. https://api.semanticscholar.org/CorpusID:273229168.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *ArXiv*, abs/2110.14168, 2021. https://api.semanticscholar.org/CorpusID:239998651.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bing-Li Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dong-Li Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Jiong Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia,

Mingchuan Zhang, Minghua Zhang, M. Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, Ruiqi Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shao-Kang Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wen-Xia Yu, Wentao Zhang, Wangding Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xi aokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyu Jin, Xi-Cheng Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yi Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yu-Jing Zou, Yujia He, Yunfan Xiong, Yu-Wei Luo, Yu mei You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yao Li, Yi Zheng, Yuchen Zhu, Yunxiang Ma, Ying Tang, Yukun Zha, Yuting Yan, Zehui Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhen guo Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zi-An Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *ArXiv*, abs/2501.12948, 2025. https://api.semanticscholar.org/CorpusID:275789950.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aur'elien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab A. AlBadawy, E I Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriele Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guanglong Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Ju-Qing Jia, Kalyan Vasuden Alwala, K. Upasani, Kate Plawiak, Keqian Li, Kenneth Heafield, Kevin R. Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuen ley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Ma hesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melissa Hall Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Niko lay Bashlykov, Nikolay Bogoychev, Niladri S. Chatterji, Olivier Duchenne, Onur cCelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasić, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ron nie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sa hana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Chandra Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stéphane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Vir ginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whit ney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yiqian Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zhengxu Yan, Zhengxing Chen, Zoe Papakipos, Aaditya K. Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adi Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Po-Yao (Bernie) Huang, Beth Loyd, Beto de Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Shang-Wen Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood,

Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco (Paco) Guzmán, Frank J. Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Han Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kaixing(Kai) Wu, U KamHou, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, A Lavender, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthias Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Mun ish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navy ata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pe dro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollár, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, S. Yu. Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Kumar Gupta, Sung-Bae Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Andrei Poenaru, Vlad T. Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xia Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The Llama 3 Herd of Models, 2024. https://api.semanticscholar.org/CorpusID:271571434.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or Herding? Reward Model Ensembles Mitigate but do not Eliminate Reward Hacking. *arXiv preprint arXiv:2312.09244*, 2023.

Aparna Elangovan, Lei Xu, Jongwoo Ko, Mahsa Elyasi, Ling Liu, Sravan Babu Bodapati, and Dan Roth. Beyond correlation: The impact of human uncertainty in measuring the effectiveness of automatic evaluation and LLM-as-a-judge. In *The Thirteenth International Conference on Learning Representations*, 2025. https://openreview.net/forum?id=E8gYIrbP00.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph Gonzalez, and Ion Stoica. How to Evaluate Reward Models for RLHF. *ArXiv*, abs/2410.14872, 2024. https://api.semanticscholar.org/CorpusID:273502060.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR. https://proceedings.mlr.press/v48/gal16.html.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *ArXiv*, abs/2104.08821, 2021. https://api.semanticscholar.org/CorpusID:233296292.

Luke Guerdan, Solon Barocas, Kenneth Holstein, Hanna Wallach, Zhiwei Steven Wu, and Alexandra Chouldechova. Validating llm-as-a-judge systems under rating indeterminacy. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Stephane Hatgis-Kessell, Logan Mondal Bhamidipaty, and Emma Brunskill. Repairing Reward Functions with Feedback to Mitigate Reward Hacking, 2025. https://api.semanticscholar.org/CorpusID:282102369.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and

Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. *ArXiv*, abs/2103.03874, 2021. https://api.semanticscholar.org/CorpusID:232134851.

Jian Hu, Xibin Wu, Weixun Wang, Songlin Jiang, Dehao Zhang, Yu Cao, OpenLLMAI Team, Netease Fuxi, AI Lab, and Alibaba Group. OpenRLHF: An Easy-to-use, Scalable and High-performance RLHF Framework. *ArXiv*, abs/2405.11143, 2024. https://api.semanticscholar.org/CorpusID:269921667.

Sukai Huang, Shu-Wei Liu, Nir Lipovetzky, and Trevor Cohn. The Dark Side of Rich Rewards: Understanding and Mitigating Noise in VLM Rewards, 2024. https://api.semanticscholar.org/CorpusID:272832041.

Gareth M. James, Daniela M. Witten, Trevor J. Hastie, and Robert Tibshirani. An Introduction to Statistical Learning. *Springer Texts in Statistics*, 2013. https://api.semanticscholar.org/CorpusID:62973643.

Nathan Lambert, Jacob Daniel Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hanna Hajishirzi. TÜLU 3: Pushing Frontiers in Open Language Model Post-Training. *ArXiv*, abs/2411.15124, 2024a. https://api.semanticscholar.org/CorpusID:274192505.

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. RewardBench: Evaluating Reward Models for Language Modeling. *ArXiv*, abs/2403.13787, 2024b. https://api.semanticscholar.org/CorpusID:268537409.

Zi Lin, Sheng Shen, Jingbo Shang, Jason Weston, and Yixin Nie. Learning to Solve and Verify: A Self-Play Framework for Code and Test Generation, 2025. https://arxiv.org/abs/2502.14948.

Chris Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. *ArXiv*, abs/2410.18451, 2024a. https://api.semanticscholar.org/CorpusID:273549327.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, et al. Skywork-Reward-V2: Scaling Preference Data Curation via Human-AI Synergy. *arXiv preprint arXiv:2507.01352*, 2025.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. RM-Bench: Benchmarking Reward Models of Language Models with Subtlety and Style. *arXiv preprint arXiv:2410.16184*, 2024b.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Daniel Morrison, Noah A. Smith, Hanna Hajishirzi, and Nathan Lambert. Rewardbench 2: Advancing reward model evaluation. In *The Fourteenth International Conference on Learning Representations*, 2025. https://api.semanticscholar.org/CorpusID:279119102.

Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. InfoRM: Mitigating Reward Hacking in RLHF via Information-Theoretic Reward Modeling. *Advances in Neural Information Processing Systems 37*, 2024. https://api.semanticscholar.org/CorpusID:267657799.

Ramon E Moore. *Interval analysis*. Prentice-Hall, 1966.

Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. Ray: a distributed framework for emerging AI applications. In *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*, OSDI'18, page 561–577, USA, 2018. USENIX Association. ISBN 9781931971478.

Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D. Lee, and Sanjeev Arora. What Makes a Reward Model a Good Teacher? An Optimization Perspective. *ArXiv*, abs/2503.15477, 2025. https://api.semanticscholar.org/CorpusID:277112967.

Jon Saad-Falcon, Estefany Kelly Buchanan, Mayee F. Chen, Tzu-Heng Huang, Brendan McLaughlin, Tanvir Bhathal, Shang Zhu, Ben Athiwaratkun, Frederic Sala, Scott W. Linderman, Azalia Mirhoseini, and Christopher Ré. Shrinking the Generation-Verification Gap with Weak Verifiers. *ArXiv*, abs/2506.18203, 2025. https://api.semanticscholar.org/CorpusID:280000478.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *ArXiv*, abs/1707.06347, 2017. https://api.semanticscholar.org/CorpusID:28695052.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. https://arxiv.org/abs/2402.03300.

Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. Distributional Preference Learning: Understanding and Accounting for Hidden Context in RLHF. In *International Conference on Learning Representations*, 2024. https://api.semanticscholar.org/CorpusID:266191810.

Hao Sun, Yunyi Shen, and Jean-François Ton. Rethinking Bradley-Terry Models in Preference-Based Reward Modeling: Foundations, Theory, and Alternatives. *ArXiv*, abs/2411.04991, 2024. https://api.semanticscholar.org/CorpusID:273877679.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct Distillation of LM Alignment. *ArXiv*, abs/2310.16944, 2023. https://api.semanticscholar.org/CorpusID:264490502.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. Checklists Are Better Than Reward Models For Aligning Language Models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. https://arxiv.org/abs/2507.18624.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10582–10592, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.620. https://aclanthology.org/2024.findings-emnlp.620/.

Ronald J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 8:229–256, 2004. https://api.semanticscholar.org/CorpusID:2332513.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing Hidden States Enables Learning Generalizable Reward Model for LLMs. *ArXiv*, abs/2406.10216, 2024. https://api.semanticscholar.org/CorpusID:270521260.

Zonglin Yang, Zhexuan Gu, Houduo Qi, and Yancheng Yuan. Accelerating RLHF Training with Reward Variance Increase. *ArXiv*, abs/2505.23247, 2025. https://api.semanticscholar.org/CorpusID:278996019.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild. In *The Twelfth International Conference on Learning Representations*, 2024. https://openreview.net/forum?id=Bl8u7ZRlbM.

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

A Proof of Proposition 2.1 (Accuracy is the weighted sum of discriminative ability and specificity)

Accuracy is

$$\begin{aligned}
&\mathbb{E}_{a,b}[\mathbb{1}[\text{sgn}(r_x(a) - r_x(b)) = \text{sgn}(u_x(a) - u_x(b))] \\
&= P\left(\text{sgn}(r_x(a) - r_x(b)) = \text{sgn}(u_x(a) - u_x(b))\right) \\
&= P\left[r_x(a) = r_x(b), u_x(a) = u_x(b)\right] \\
&\quad + P\left[r_x(a) > r_x(b), u_x(a) > u_x(b)\right] \\
&\quad + P\left[r_x(a) < r_x(b), u_x(a) < u_x(b)\right]
\end{aligned}$$

First term:

$$\begin{aligned}
&P\left[r_x(a) = r_x(b), u_x(a) = u_x(b)\right] \\
&= (1 - P[r_x(a) \neq r_x(b) \mid u_x(a) = u_x(b)) \times P\left[u_x(a) = u_x(b)\right]
\end{aligned}$$

Second and third terms:

$$\begin{aligned}
&P\left[r_x(a) > r_x(b), u_x(a) > u_x(b)\right] + P\left[r_x(a) < r_x(b), u_x(a) < u_x(b)\right] \\
&= P\left[r_x(a) > r_x(b) \mid u_x(a) > u_x(b)\right] P(u_x(a) > u_x(b)) \;\; + P\left[r_x(a) < r_x(b) \mid u_x(a) < u_x(b)\right] P(u_x(a) < u_x(b)).
\end{aligned}$$

Then, accuracy is

$$\begin{aligned}
&(1 - P[r_x(a) \neq r_x(b) \mid u_x(a) = u_x(b)) \\
&\quad \times P\left[u_x(a) = u_x(b)\right] && \textit{(specificity)} \\
&+ P\left[r_x(a) > r_x(b) \mid u_x(a) > u_x(b)\right] P(u_x(a) > u_x(b))] \\
&+ P\left[r_x(a) < r_x(b) \mid u_x(a) < u_x(b)\right] P(u_x(a) < u_x(b))] && \textit{(discriminative ability)}
\end{aligned}$$

B Proof of Proposition 2.5 (The oversensitivity of a binary-discretized reward model is $\max(0, \frac{1}{2} - 2((\tau_x - \phi_x(u_x(a)))/s_x)^2)$)

Proof For sufficiently small ϵ ($\epsilon < s_x$), consider that $|\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b))| \iff |\text{disc}_x(r_x(a)) - \text{disc}_x(r_x(b))| > \epsilon$. Therefore, $P(|\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b))| + \epsilon \mid u_x(a) = u_x(b)) = P(\text{disc}_x(r_x(a)) \neq \text{disc}_x(r_x(b)) \mid u_x(a) = u_x(b))$

For our discretized (binary) reward to retain *perfect discriminative ability*, this implies that there are exactly two unique values of utility for this task, meaning this task has a *binary* notion of correctness. In this setting:

$$P(\text{disc}_x(r_x(a)) \neq \text{disc}_x(r_x(b))) \quad = \quad P(\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b))) \quad + \quad P(\text{disc}_x(r_x(b)) > \text{disc}_x(r_x(a))).$$

Conditioned on $u_x(a) = u_x(b)$, these two orderings are equally likely. We can compute the first and double it:

$$\begin{aligned}
&P(\text{disc}_x(r_x(a)) > \text{disc}_x(r_x(b))) = P(r_x(a) > \tau_x \text{ and } r_x(b) \leq \tau_x) = \\
&P(\phi_x(u_x(a)) + \eta_x(a) > \tau_x \quad \text{ and } \quad \phi_x(u_x(b)) + \eta_x(b) \leq \tau_x) = \\
&P(\phi_x(u_x(a)) + \eta_x(a) > \tau_x) \;\; \times \;\; P(\phi_x(u_x(b)) + \eta_x(b) \leq \tau_x).
\end{aligned}$$

The final line results from the fact that the events that $\phi_x(u_x(a)) + \eta_x(a) > \tau_x$ and $\phi_x(u_x(b)) + \eta_x(b) \leq \tau_x$ are independent, conditioned on the fact that $u_x(a) = u_x(b)$.

First term:

$$
\begin{aligned}
&P(\phi_x(u_x(a)) + \eta_x(a) > \tau_x) \\
&= P\big(\text{Unif}(-s_x/2, s_x/2) > \tau_x - \phi_x(u_x(a))\big) \\
&= P\big(\text{Unif}(-1/2, 1/2) > \mathit{distance}\big) \\
&\quad (\mathit{where}\ \mathit{distance} := (\tau_x - \phi_x(u_x(a)))/s_x) = \\
&= \begin{cases} 1, & \text{if } \mathit{distance} \le -\frac{1}{2} \\ 0, & \text{if } \mathit{distance} \ge \frac{1}{2} \\ \frac{1}{2} - \mathit{distance}, & \text{otherwise.} \end{cases}
\end{aligned}
$$

Similarly, second term:

$$
\begin{aligned}
&P(\phi_x(u_x(b)) + \eta_x(b) \le \tau_x) \\
&= P\big(\text{Unif}(-1/2, 1/2) \le (\tau_x - \phi_x(u_x(b)))/s_x\big) \\
&= \begin{cases} 0, & \text{if } \mathit{distance} \le -\frac{1}{2} \\ 1, & \text{if } \mathit{distance} \ge \frac{1}{2} \\ \mathit{distance} + \frac{1}{2}, & \text{otherwise.} \end{cases}
\end{aligned}
$$

Given that $u_x(a) = u_x(b)$, we multiply these two terms and then double the result to account for both orderings:

$$
\begin{aligned}
&2P(\phi_x(u_x(a)) + \eta_x(a) > \tau_x) \\
&\qquad \times P(\phi_x(u_x(b)) + \eta_x(b) \le \tau_x) \\
&= 2 \begin{cases} 0, & \text{if } \mathit{distance} \le -\frac{1}{2} \\ 0, & \text{if } \mathit{distance} \ge \frac{1}{2} \\ 1/4 - (\mathit{distance})^2, & \text{otherwise.} \end{cases} \\
&= \max(0, 1/2 - 2(\mathit{distance})^2) \\
&= \max\big(0, 1/2 - 2((\tau_x - \phi_x(u_x(a)))/s_x)^2\big)
\end{aligned}
$$

Then $\text{Spec}_{\text{disc}} = \min\big(1, 2((\tau_x - \phi_x(u_x(a)))/s_x)^2 - 1/2\big)$.

C Ablation of Number of Dropout Samples

Section 3 introduces our algorithm for reward clustering, which uses MC dropout to estimate the predictive variance of a reward model. In Table 3, we show results from ablating the number of samples taken via MC dropout for *Skywork V1*. We find that the number of samples has surprisingly little impact on the intrinsic efficacy of our reward discretization method. We observe identical patterns with the other reward models used in this paper.

Method	*Proposed* Metrics			Ties		
	Avg.	Discrim.	Specif.	Overall	Acc.	Correctness
			Skywork V1			
$T = 3$	70.8 ± 0.8	98.0 ± 0.4	43.6 ± 1.6	0.729 ± 0.029	0.882 ± 0.025	0.500 ± 0.048
★ $\boldsymbol{T = 4}$	70.9 ± 0.8	98.2 ± 0.4	43.6 ± 1.6	0.729 ± 0.029	0.882 ± 0.025	0.500 ± 0.048
$T = 8$	70.7 ± 0.8	98.1 ± 0.4	43.3 ± 1.6	0.724 ± 0.031	0.873 ± 0.028	0.500 ± 0.048
$T = 12$	70.9 ± 0.8	98.2 ± 0.4	43.6 ± 1.6	0.729 ± 0.029	0.882 ± 0.025	0.500 ± 0.048

Table 3 We find the effectiveness of reward clustering via MC dropout is robust to the number of dropout samples used. We report reward clustering performance for Skywork V1. ★ indicates the default setting ($T = 4$).

D Discretization still improves reward models if we use a Gaussian noise mode

D.1 A relaxed noise model: Gaussian-distributed rewards

In Section 2.1 we assumed bounded uniform noise $\eta_x(a) \sim \mathrm{Unif}(-s_x/2, s_x/2)$, leading to uniform-distributed rewards. These hard noise limits guaranteed perfect discriminative ability of the raw reward model; without it, the analysis in §2.2–2.4 no longer applies. We now relax this assumption to *Gaussian* noise. This means that the raw reward model can no longer be perfect. Here we assume that true utility is binary, as in §2.2, but these concepts readily generalize to multi-level utility functions.

Assumption. For a fixed prompt x, we model the reward as $r_x(a) = \phi_x(u_x(a)) + \eta_x(a)$, where $\phi_x(v) = s_x v + d_x$ and $\eta_x(a) \sim \mathcal{N}(0, \sigma_x^2)$. The per-prompt noise is homoskedastic: σ_x and s_x are shared across utility classes. We define the per-prompt *signal-to-noise ratio* as $\rho_x := s_x/\sigma_x$; this determines the reward model's accuracy. For convenience, we drop the subscript x and use Φ and φ for the Gaussian CDF and density, respectively.

D.2 Discriminative ability and oversensitivity under Gaussian noise

Proposition D.1. *Under the homoskedastic Gaussian model with binary utility,*

$$D_{\mathrm{raw}}(\epsilon) = \big(r_x(a) - r_x(b) > \epsilon \mid u_x(a) > u_x(b)\big) = \Phi\left(\rho_x(1-\epsilon)/\sqrt{2}\right).$$

Proof. For some a, b where $u_x(a) > u_x(b)$, then $r_x(a) - r_x(b) = s_x + \eta_x(a) - \eta_x(b)$, where $\eta_x(b) - \eta_x(a) \sim \mathcal{N}(0, 2\sigma_x^2)$. Thus $D_{\mathrm{raw}}(\epsilon) = P\big(\eta_x(b) - \eta_x(a) < s_x - \epsilon\big) = \Phi\big((s_x - \epsilon)/(\sigma_x\sqrt{2})\big)$, which equals $\Phi(\rho_x(1-\epsilon)/\sqrt{2})$.

As in subsection 2.4, we measure reward in *utility units* to be able to directly compare between raw and discrete rewards. The noise within each utility class here is now $\eta_x/s_x \sim \mathcal{N}(0, 1/\rho_x^2)$.

Proposition D.2. *In utility units, for any $\epsilon \geq 0$,*

$$\mathrm{Spec}_{\mathrm{raw}}(\epsilon) = P\big(|r_x(a) - r_x(b)|/s_x < \epsilon \;\big|\; u_x(a) = u_x(b)\big) = 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1.$$

Proof. Conditional on $u_x(a) = u_x(b)$, $(r_x(a) - r_x(b))/s_x = (\eta_x(a) - \eta_x(b))/s_x \sim \mathcal{N}(0, 2/\rho_x^2)$. Hence $P(|\mathcal{N}(0, 2/\rho_x^2)| < \epsilon) = P(\mathcal{N}(0, 2/\rho_x^2) < \epsilon) - P(\mathcal{N}(0, 2/\rho_x^2) < -\epsilon) = \Phi(\epsilon/\sqrt{2/\rho_x^2}) - \Phi(-\epsilon/\sqrt{2/\rho_x^2}) = \Phi(\epsilon/\sqrt{2/\rho_x^2}) - (1 - \Phi(\epsilon/\sqrt{2/\rho_x^2})) = 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1$.

D.3 Discretization preserves most discriminative ability and substantially reduces oversensitivity

Proposition D.3. *For binary utility, the discriminative ability of the discretized reward, given two adjacent utility classes, is maximized at $\tau_x := (\phi_x(u_x(a)) + \phi_x(u_x(b)))/2$. Discriminative ability under this threshold is*

$$D_{\mathrm{disc}}(\epsilon) = P\big(\mathrm{disc}_x(r_x(a)) > \mathrm{disc}_x(r_x(b)) \mid u_x(a) > u_x(b)\big) = \Phi(\rho_x/2)^2.$$

Proof. Assume that $u_x(a) = 1, u_x(b) = 0$. Since the events $\{r_x(a) > \tau_x\}$ and $\{r_x(b) \leq \tau_x\}$ are independent:

$$P\big(\mathrm{disc}_x(r_x(a)) > \mathrm{disc}_x(r_x(b)) \mid u_x(a) > u_x(b)\big) = \Phi\left(\tfrac{\phi_x(1) - \tau_x}{\sigma_x}\right)\Phi\left(\tfrac{\tau_x - \phi_x(0)}{\sigma_x}\right).$$

We wish to find a value of τ_x that maximizes this probability. Let us denote $z := (\tau_x - \phi_x(0))/\sigma_x$ (this is the continuous analog of *distance* used in Appendix B) . Then we can rewrite $(\phi_x(1) - \tau_x)/\sigma_x$ as $\rho_x - z$. This in turn allows us to rewrite $P\big(\mathrm{disc}_x(r_x(a)) > \mathrm{disc}_x(r_x(b)) \mid u_x(a) > u_x(b)\big)$ as $\Phi(\rho_x - z)\Phi(z)$.

Now let $f(z) := \Phi(\rho_x - z)\Phi(z)$. We can maximize this function by setting its derivative to zero (note that φ refers to the derivative of Φ):

$$f'(z) = -\varphi(\rho_x - z)\Phi(z) + \Phi(\rho_x - z)\varphi(z) = 0 \iff \tfrac{\varphi(z)}{\varphi(\rho_x - z))} = \tfrac{\Phi(z)}{\Phi(\rho_x - z)} \iff \tfrac{\varphi(z)}{\Phi(z)} = \tfrac{\varphi(\rho_x - z)}{\Phi(\rho_x - z)},$$

which holds at $z = \rho_x/2$. Then the optimizer is $\tau_x = \phi_x(0) + s_x/2 = (\phi_x(0) + \phi_x(1))/2$, and $D_{\mathrm{disc}}(\epsilon) = \Phi(\rho_x/2)^2$.

Proposition D.4. *At the threshold τ_x of Proposition D.3, for any $0 \le \epsilon < 1$ and binary utility:*

$$\mathrm{Spec}_{\mathrm{disc}}(\epsilon) = 1 - P\big(|\mathrm{disc}_x(r_x(a)) - \mathrm{disc}_x(r_x(b))| > \epsilon \;\big|\; u_x(a) = u_x(b)\big) = 1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2).$$

Proof. Since disc_x is binary-valued, then for $0 \le \epsilon < 1$, $|\mathrm{disc}_x(r_x(a)) - \mathrm{disc}_x(r_x(b))| > \epsilon \iff \mathrm{disc}_x(r_x(a)) \neq \mathrm{disc}_x(r_x(b))$.

Let $z_u := (\tau_x - \phi_x(u))/\sigma_x$, representing the *distance* from any given utility class. Conditional on $u_x(a) = u_x(b)$, $\mathrm{disc}_x(r_x(a))$ and $\mathrm{disc}_x(r_x(b))$ are independent. Then:

$$P\big(\mathrm{disc}_x(r_x(a)) \neq \mathrm{disc}_x(r_x(b)) \mid u_x(a) = u_x(b)\big) = 2\big(1 - \Phi(z_u)\big)\Phi(z_u) = 2\Phi(-z_u)\Phi(z_u).$$

Proposition D.3 shows discriminative ability is maximized at $\tau_x = \phi_x(0) + s_x/2 = s_x/2 + d_x = (\phi_x(0) + \phi_x(1))/2$. Then, the distance of this threshold to $u_x(a) = 0$ is $z_0 = \rho_x/2$ and the distance to $u_x(a) = 1$ is $z_1 = -\rho_x/2$. Regardless of the utility class u_x, we then obtain the same oversensitivity: $2\Phi(\rho_x/2)\Phi(-\rho_x/2)$. Then $\mathrm{Spec}_{\mathrm{disc}}(\epsilon) = 1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2)$, taking the complement of oversensitivity.

Remark D.5 (the shape difference). The oversensitivity of the discretized reward does not depend on ϵ: it has a flat value of $2\Phi(\rho_x/2)\Phi(-\rho_x/2)$ across all $\epsilon \in [0, 1)$. The raw oversensitivity depends on ϵ: $2\Phi(-\epsilon\rho_x/\sqrt{2})$ starts at 1 and decays as ϵ increases. This shape difference is key for understanding why discretization is better (in Theorem D.7).

Remark D.6 (choosing the threshold). We use the midpoint discretization threshold $\tau_x = (\phi_x(0) + \phi_x(1))/2$ throughout. This threshold maximizes discriminative ability (Proposition D.3). It does not globally minimize oversensitivity, but rather it yields a favorable tradeoff, as we will show in Theorem D.9.

D.4 The discretization tradeoff

Combining Propositions D.1–D.4, discretization reduces the discriminative ability from $\Phi(\rho_x/\sqrt{2})$ to $\Phi(\rho_x(1-\epsilon)/\sqrt{2})$, and discretization changes specificity from $2\Phi(\epsilon\rho_x/\sqrt{2}) - 1$ to $1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2)$. The discretized reward 's discriminative ability and specificity are both *constant* for $\epsilon \in [0, 1)$. In contrast, the raw reward sees its discriminative ability $\Phi(\rho_x(1-\epsilon)/\sqrt{2})$ decline while its specificity $2\Phi(\epsilon\rho_x/\sqrt{2}) - 1$ rises from 0 as ϵ grows.

Theorem D.7. *For every signal-to-noise ratio $\rho_x > 0$ and every tolerance $\epsilon \in [0, 1/\sqrt{2}]$, $\mathrm{Spec}_{\mathrm{disc}}(\epsilon) > \mathrm{Spec}_{\mathrm{raw}}(\epsilon)$ (discretization improves the specificity of the reward model).*

Proof. Propositions D.2 and D.4 give $\mathrm{Spec}_{\mathrm{raw}}(\epsilon) = 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1$ and $\mathrm{Spec}_{\mathrm{disc}}(\epsilon) = 1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2)$. Using $\Phi(\epsilon\rho_x/\sqrt{2}) = 1 - \Phi(-\epsilon\rho_x/\sqrt{2})$, then

$$\begin{aligned}\mathrm{Spec}_{\mathrm{disc}}(\epsilon) - \mathrm{Spec}_{\mathrm{raw}}(\epsilon) &= 1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2) - 2(1 - \Phi(-\epsilon\rho_x/\sqrt{2})) + 1 \\ &= 2\big[\Phi(-\epsilon\rho_x/\sqrt{2}) - \Phi(\rho_x/2)\Phi(-\rho_x/2)\big],\end{aligned}$$

Because ρ_x is positive, $\epsilon < 1/\sqrt{2} \implies \epsilon/\sqrt{2} < 1/2 \implies \Phi(-\epsilon\rho_x/\sqrt{2}) > \Phi(-\rho_x/2)$. Because $\Phi(\rho_x/2) < 1$, $\Phi(-\epsilon\rho_x/\sqrt{2}) > \Phi(\rho_x/2)\Phi(-\rho_x/2)$, and then $\mathrm{Spec}_{\mathrm{disc}}(\epsilon) - \mathrm{Spec}_{\mathrm{raw}}(\epsilon) > 0$.

This means that for every ϵ below $1/\sqrt{2}$ (in utility units) and at every signal-to-noise ratio, discretization is strictly more specific than the raw reward. In practice, this means that the specificity improves as long as we declare two reward values as meaningfully different as long as their difference in rewards exceeds at least $\sim 70\%$ of the proportional difference in their true utility values. This covers most of the plausible range of ϵ that a practitioner might consider.

Weighing the two axes at a common tolerance. The success condition isolates specificity. To support the stronger claim that discretization *preserves most discriminative ability while* improving specificity, consider their average at a common tolerance ϵ. The raw discriminative ability is $D_{\mathrm{raw}}(\epsilon) = \Phi(\rho_x(1-\epsilon)/\sqrt{2})$ (Proposition D.1); the discretized discriminative ability is $D_{\mathrm{disc}}(\epsilon) = \Phi(\rho_x/2)^2$ (Proposition D.3). As in Theorem 2.7, we consider the single combined score $T_r(\epsilon) := (D_r(\epsilon) + \mathrm{Spec}_r(\epsilon))/2$.

Proposition D.8. *Consider the averaged tradeoff between specificity and discrminative ability:* $T(\epsilon) = (\mathrm{Spec}(\epsilon) + D(\epsilon))/2$. *The gap between this value computed for a discretized reward model and the raw reward models is minimized at the worst-case tolerance* $\epsilon = \min\{\frac{1}{2} + \frac{2\log 2}{\rho_x^2}, 1\}$.

Proof. Using $D_{\text{disc}}(\epsilon) = \Phi(\rho_x/2)^2$ from Proposition D.3, $\mathrm{Spec}_{\text{disc}}(\epsilon) = 1 - 2\Phi(\rho_x/2)\Phi(-\rho_x/2)$ from Proposition D.4, and $\Phi(-\rho_x/2) = 1 - \Phi(\rho_x/2)$, then

$$T_{\text{disc}} = (D_{\text{disc}} + \mathrm{Spec}_{\text{disc}})/2 \;=\; \big(\Phi(\rho_x/2)^2 \;+\; 1 - 2\Phi(\rho_x/2)(\Phi(-\rho_x/2))\big)/2 \;=\; \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) + \tfrac{1}{2}$$

Using $D_{\text{raw}}(\epsilon) = \Phi\left(\rho_x(1-\epsilon)/\sqrt{2}\right)$ (Proposition D.1) and $\mathrm{Spec}_{\text{raw}}(\epsilon) = 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1$ (Proposition D.2),

$$T_{\text{raw}}(\epsilon) = (D_{\text{raw}} + \mathrm{Spec}_{\text{raw}})/2 \;=\; \big(\Phi(\rho_x(1-\epsilon)/\sqrt{2}) + 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1\big)/2$$

Denote $T_\Delta(\epsilon) = \; T_{\text{disc}}(\epsilon) - T_{\text{raw}}(\epsilon) \;=\; \frac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) + \frac{1}{2} - \big(\big(\Phi(\rho_x(1-\epsilon)/\sqrt{2}) + 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1\big)/2\big)$. First, we can ask where T_Δ is minimal. Setting the derivative to zero:

$$\begin{aligned}
&T'_\Delta(\epsilon) = \tfrac{d}{d\epsilon}\big((-\Phi(\rho_x(1-\epsilon)/\sqrt{2}) - 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1)/2\big) = 0 \\
&\iff \tfrac{1}{2}\big(\tfrac{\rho_x}{\sqrt{2}}\varphi(\rho_x(1-\epsilon)/\sqrt{2}) - 2\tfrac{\rho_x}{\sqrt{2}}\varphi(\epsilon\rho_x/\sqrt{2})\big) = 0 \\
&\iff \tfrac{\rho_x}{2\sqrt{2}}\big(\varphi(\rho_x(1-\epsilon)/\sqrt{2}) - 2\varphi(\epsilon\rho_x/\sqrt{2})\big) = 0 \\
&\implies 2\varphi(\epsilon\rho_x/\sqrt{2}) = \varphi(\rho_x(1-\epsilon)/\sqrt{2}) \\
&\implies 2\tfrac{1}{\sqrt{2\pi}}\exp\left(-\tfrac{1}{2}(\epsilon\rho/\sqrt{2})^2\right) = \tfrac{1}{\sqrt{2\pi}}\exp\left(-\tfrac{1}{2}(\rho_x(1-\epsilon)/\sqrt{2})^2\right)
\end{aligned}$$

Taking the logarithm of both sides, we find a unique minimizer:

$$\begin{aligned}
&\implies \log 2 - \log(\sqrt{2\pi}) - \tfrac{1}{2}(\epsilon\rho/\sqrt{2})^2 = -\log(\sqrt{2\pi}) - \tfrac{1}{2}(\rho_x(1-\epsilon)/\sqrt{2})^2 \\
&\implies \tfrac{1}{2}(\epsilon\rho/\sqrt{2})^2 - \tfrac{1}{2}(\rho_x(1-\epsilon)/\sqrt{2})^2 = \log 2 \\
&\implies \tfrac{1}{4}(2\epsilon - 1)\rho_x^2 = \log 2 \implies \epsilon^* = \tfrac{2\log 2}{\rho_x^2} + \tfrac{1}{2}.
\end{aligned}$$

We know the gradient is zero at this value is ϵ^*. If $\epsilon^* \in [0, 1)$, then we have found an optimum on $[0, 1)$. Since this may not be the case (e.g. $\rho_x < 2\sqrt{\log 2}$), we can consider the sign of the gradient.

$$\begin{aligned}
&\mathrm{sgn}(T'_\Delta(\epsilon)) = \mathrm{sgn}\Big(\tfrac{\rho_x}{2\sqrt{2}}\big(\varphi(\rho_x(1-\epsilon)/\sqrt{2}) - 2\varphi(\epsilon\rho_x/\sqrt{2})\big)\Big) \\
&= \mathrm{sgn}\big(\varphi(\rho_x(1-\epsilon)/\sqrt{2}) - 2\varphi(\epsilon\rho_x/\sqrt{2})\big)
\end{aligned}$$

Since both terms are positive, the inequality is preserved under the logarithm:

$$\begin{aligned}
&= \mathrm{sgn}\big(\log\varphi(\rho_x(1-\epsilon)/\sqrt{2}) - \log(2\varphi(\epsilon\rho_x/\sqrt{2}))\big) \\
&= \mathrm{sgn}\Big[\log\big(\tfrac{1}{\sqrt{2\pi}}\exp\big(-\tfrac{1}{2}(\rho_x(1-\epsilon)/\sqrt{2})^2\big)\big) - \log\big(\tfrac{1}{\sqrt{2\pi}}\exp\big(-\tfrac{1}{2}(\epsilon\rho_x/\sqrt{2})^2\big)\big) - \log 2\Big] \\
&= \mathrm{sgn}\big(-\tfrac{1}{2}(\rho_x(1-\epsilon)/\sqrt{2})^2 - (-\tfrac{1}{2}(\epsilon\rho_x/\sqrt{2})^2) - \log 2\big) \\
&= \mathrm{sgn}\big(\tfrac{1}{4}\rho_x^2(2\epsilon - 1) - \log 2\big)
\end{aligned}$$

The gradient then changes sign at $\epsilon = \frac{1}{2} + \frac{2\log 2}{\rho_x^2}$; the gradient is negative for $\epsilon < \epsilon^*$ and positive for $\epsilon > \epsilon^*$. Therefore, we know that ϵ^* is the global minimum of T_Δ, and we know that if $\epsilon > 1$, then $T_\Delta(1)$ is the minimum of T_Δ over $[0, 1)$. As we reduce $\epsilon \to 0$, the gap between discretized and raw rewards grows further.

Theorem D.9. *The specificity gain exceeds the discriminative cost: for every* $\rho_x > 0$ *and every tolerance* $\epsilon \in [0, 1)$: $\;\; T_{\text{disc}}(\epsilon) - T_{\text{raw}}(\epsilon) > 0$. *As* $\rho_x \to \infty$, *the net benefit approaches zero: when the raw reward itself approaches perfection, discretization is unnecessary.*

Proof. Proposition D.8 shows that $T_{\text{disc}}(\epsilon) - T_{\text{raw}}(\epsilon)$ is minimized at $\epsilon = \min(1, \epsilon^*)$ where $\epsilon^* = \frac{1}{2} + \frac{2\log 2}{\rho_x^2}$. If we can show that $T_\Delta\big(\min\{1, \epsilon^*\}\big) > 0$ for every $\rho_x > 0$; then T_Δ is positive for all values of $\epsilon \in [0, 1)$.

Denote the worst-case value by $\Delta(\rho_x) := T_\Delta\big(\min\{1, \epsilon^*\}\big)$. We will divide our analysis by ranges of ρ_x. Our three cases are $0 < \rho_x \le 2\sqrt{\log 2}$, $2\sqrt{\log 2} < \rho_x < 4.2$, and $\rho_x \ge 4.2$.

Case 1: $0 < \rho_x \le 2\sqrt{\log 2}$. Then, $\epsilon^* \ge 1$, so T_Δ is strictly decreasing on all of $[0,1)$: $T_\Delta(\epsilon) > T_\Delta(1)$. Since T_Δ is continuous at 1, it is therefore sufficient to show that $T_\Delta(1) \ge 0$.

$$T_\Delta(1) = \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) - \Phi(\rho_x/\sqrt{2}) + \tfrac{3}{4}.$$

Differentiating, and using $\varphi(\rho_x/\sqrt{2}) = \varphi(\rho_x/2)e^{-\rho_x^2/8}$,

$$\begin{aligned}
\Delta'(\rho_x) &= \tfrac{3}{2}\Phi(\tfrac{\rho_x}{2})\varphi(\tfrac{\rho_x}{2}) - \tfrac{1}{2}\varphi(\tfrac{\rho_x}{2}) - \tfrac{\sqrt{2}}{2}\varphi(\tfrac{\rho_x}{\sqrt{2}}) \\
&= \tfrac{3}{2}\Phi(\tfrac{\rho_x}{2})\varphi(\tfrac{\rho_x}{2}) - \tfrac{1}{2}\varphi(\tfrac{\rho_x}{2}) - \tfrac{\sqrt{2}}{2}\big(-\tfrac{1}{\sqrt{2\pi}}\exp\left(-(\rho_x/\sqrt{2})^2/2\right)\big) \\
&= \tfrac{3}{2}\Phi(\tfrac{\rho_x}{2})\varphi(\tfrac{\rho_x}{2}) - \tfrac{1}{2}\varphi(\tfrac{\rho_x}{2}) - \tfrac{\sqrt{2}}{2}\big(-\tfrac{1}{\sqrt{2\pi}}\exp\left(-(\rho_x/2)^2/2\right)\exp\left(-\rho_x^2/8\right)\big) \\
&= \tfrac{3}{2}\Phi(\tfrac{\rho_x}{2})\varphi(\tfrac{\rho_x}{2}) - \tfrac{1}{2}\varphi(\tfrac{\rho_x}{2}) - \tfrac{\sqrt{2}}{2}\varphi(\tfrac{\rho_x}{2})\exp\left(-\rho_x^2/8\right)\big) \\
&= \tfrac{1}{2}\varphi(\tfrac{\rho_x}{2})\big(3\Phi(\tfrac{\rho_x}{2}) - 1 - \sqrt{2}\exp\left(-\rho_x^2/8\right)\big)
\end{aligned}$$

$\frac{1}{2}\varphi(\rho_x/2)$ is always positive. The other term $A(\rho_x) = 3\Phi(\rho_x/2) - 1 - \sqrt{2}e^{-\rho_x^2/8}$ is strictly increasing in ρ_x: $\frac{d}{d\rho_x}\big(A(\rho_x)\big) = \frac{3}{2}\varphi(\rho_x/2) + \frac{\sqrt{2}}{4}\rho_x e^{-\rho_x^2/8} > 0$ for $\rho_x > 0$. We know $A(0) = \frac{1}{2} - \sqrt{2} < 0$ and $A(2\sqrt{\log 2}) = 3\Phi(\sqrt{\log 2}) - 2 > 0$, so its zero must lie between these two arguments of A. We can numerically find further that the zero (and, therefore, the minimizer of $\Delta(\rho_x)$) is between $\rho_x = 1.204$ and $\rho_x = 1.205$. At both endpoints here, where $\Delta(\rho_x) > 0.012$, and on this interval, $\frac{1}{2}\varphi(\rho_x/2) < 0.1665$, while $-0.0006 < A(\rho_x) < 0.0003$. A lower-bound value for $\Delta'(\rho_x)$ (most-negative possible slope) is -10^{-4}. Then, by the Mean Value Theorem, the minimum value of $\Delta(\rho_x)$ must be at least $0.012 - (1.205 - 1.204)\cdot 10^{-4}$, so $\Delta(\rho_x)$ is always positive with a minimum value of at least 0.0119999.

Case 2: $\rho_x \ge 4.2$, where this constant is chosen for reasons explained later in this section. Here $\epsilon^* < 1$ is the minimizer.

$$\Delta(\rho_x) = \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) + \tfrac{1}{2} - \big(\Phi(\rho_x(1-\epsilon)/\sqrt{2}) + 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1\big)/2.$$

First, consider the limit as $\rho_x \to \infty$. Since $\lim_{x\to\infty}\Phi(x) = 1$, it is easy to see that $\lim_{\rho_x\to\infty}\Delta(\rho_x) = \frac{3}{2} - 1 + \frac{1}{2} - 1 = 0$. Now we consider the finite case. For convenience, we will use the complement of the Gaussian cumulative distribution function $\overline{\Phi} := 1 - \Phi$,

$$\begin{aligned}
\Delta(\rho_x) &= \tfrac{3}{2}(1-\overline{\Phi}(\rho_x/2))^2 - (1-\overline{\Phi}(\rho_x/2)) + \tfrac{1}{2} - \big((1-\overline{\Phi}(\rho_x(1-\epsilon)/\sqrt{2})) + 2(1-\overline{\Phi}(\epsilon\rho_x/\sqrt{2})) - 1\big)/2 \\
&= \tfrac{3}{2}\overline{\Phi}(\rho_x/2)^2 - 3\overline{\Phi}(\rho_x/2) + \tfrac{3}{2} - (1-\overline{\Phi}(\rho_x/2)) + \tfrac{1}{2} - \big(2 - \overline{\Phi}(\rho_x(1-\epsilon)/\sqrt{2}) - 2\overline{\Phi}(\epsilon\rho_x/\sqrt{2})\big)/2 \\
&= \tfrac{3}{2}\overline{\Phi}(\rho_x/2)^2 - 2\overline{\Phi}(\rho_x/2) + \tfrac{1}{2}\overline{\Phi}(\rho_x(1-\epsilon)/\sqrt{2}) + \overline{\Phi}(\epsilon\rho_x/\sqrt{2})
\end{aligned}$$

Substituting $\frac{1}{2} + \frac{2\log 2}{\rho_x^2}$ for ϵ:

$$= \tfrac{3}{2}\overline{\Phi}(\rho_x/2)^2 - 2\overline{\Phi}(\rho_x/2) + \tfrac{1}{2}\overline{\Phi}\Big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\Big) + \overline{\Phi}\Big(\tfrac{\rho_x}{2\sqrt{2}} + \tfrac{\sqrt{2}\log 2}{\rho_x}\Big).$$

We can eliminate the two nonnegative terms $\overline{\Phi}(\frac{\rho_x}{2\sqrt{2}} + \frac{\sqrt{2}\log 2}{\rho_x})$ and $\frac{3}{2}\overline{\Phi}(\rho_x/2)^2$:

$$\Delta(\rho_x) \ge \tfrac{1}{2}\overline{\Phi}\Big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\Big) - 2\overline{\Phi}(\rho_x/2).$$

Then we need to show that $\frac{1}{2}\overline{\Phi}\Big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\Big) - 2\overline{\Phi}(\rho_x/2) > 0$, which is the same as

$$\overline{\Phi}\big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\big)/(\overline{\Phi}(\rho_x/2)) > 4.$$

Recall that the Mills' ratio for the normal distribution requires that $t/(t^2+1) < \overline{\Phi}(t)/\varphi(t) < 1/t$. We can apply the lower bound to the numerator and the upper bound to the denominator

$$\overline{\Phi}\Big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\Big) > \frac{\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}}{\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)^2 + 1}\varphi\Big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\Big) \text{ and } \overline{\Phi}(\tfrac{\rho_x}{2}) < \tfrac{2}{\rho_x}\varphi(\rho_x/2).$$

Then, for the quantity that we want to show is greater than 4, we see a new lower bound:

$$\begin{aligned}
\frac{\overline{\Phi}\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{\overline{\Phi}(\rho_x/2)} &> \frac{\rho_x\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{2\big[\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)^2 + 1\big]} \cdot \frac{\varphi\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{\varphi(\rho_x/2)} \\
&= \frac{\frac{\rho_x^2}{2\sqrt{2}} - \sqrt{2}\log 2}{2\big[\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)^2 + 1\big]} \cdot \frac{\varphi\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{\varphi(\rho_x/2)} \\
&= \frac{\sqrt{2}(1 - \frac{4\log 2}{\rho_x^2})}{1 + \frac{16\log^2 2}{\rho_x^4} + \frac{8-8\log 2}{\rho_x^2}} \cdot \frac{\varphi\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{\varphi(\rho_x/2)}
\end{aligned}$$

Using $\varphi(t) = \frac{1}{\sqrt{2\pi}}e^{-t^2/2}$, we can simplify the second term here:

$$\begin{aligned}
\frac{\varphi\big(\frac{\rho_x}{2\sqrt{2}} - \frac{\sqrt{2}\log 2}{\rho_x}\big)}{\varphi(\rho_x/2)} &= \exp\Big(\tfrac{1}{2}\Big[(\rho_x/2)^2 - \big(\tfrac{\rho_x}{2\sqrt{2}} - \tfrac{\sqrt{2}\log 2}{\rho_x}\big)^2\Big]\Big) = \exp\Big(\tfrac{1}{2}\Big[\tfrac{\rho_x^2}{4} - \tfrac{\rho_x^2}{8} - \tfrac{2\log^2 2}{\rho_x^2} + \log 2\Big]\Big) \\
&= \exp\Big(\tfrac{1}{2}\tfrac{\rho_x^2}{8} - \tfrac{1}{2}\tfrac{2\log^2 2}{\rho_x^2} + \log 2^{\frac{1}{2}}\Big) = \sqrt{2}\exp\Big(\tfrac{\rho_x^2}{16} - \tfrac{\log^2 2}{\rho_x^2}\Big).
\end{aligned}$$

Now our lower bound is

$$\frac{\sqrt{2}(1 - \frac{4\log 2}{\rho_x^2})}{1 + \frac{16\log^2 2}{\rho_x^4} + \frac{8-8\log 2}{\rho_x^2}} \cdot \sqrt{2}\exp\Big(\tfrac{\rho_x^2}{16} - \tfrac{\log^2 2}{\rho_x^2}\Big)$$

Both terms increase as we ρ_x goes from $2\sqrt{\log 2} \to \infty$. Analytically, we can easily find values of ρ_x where this lower bound exceeds 4. One such value is $\rho_x = 4.2$, where the lower bound is 4.24. Therefore, we know $\Delta(\rho_x) > 0$ for all $\rho_x \geq 4.2$, and we will use this $\rho_x = 4.2$ as the floor of this case.

Therefore, we must consider one final case, where $2\sqrt{\log 2} < \rho_x < 4.2$. This case fails our (loose) lower bound. Since this is a bounded subset of $\mathbb{R}$, we can use interval arithmetic (Moore, 1966).

Case 3: $2\sqrt{\log 2} < \rho_x < 4.2$. We will substitute $\epsilon = \frac{1}{2} + \frac{2\log 2}{\rho_x^2}$ into our definition of $\Delta(\rho_x)$:

$$\begin{aligned}
\Delta(\rho_x) &= \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) + \tfrac{1}{2} - \big(\Phi(\rho_x(1-\epsilon)/\sqrt{2}) + 2\Phi(\epsilon\rho_x/\sqrt{2}) - 1\big)/2 \\
&= \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) + \tfrac{1}{2} - \tfrac{1}{2}\Phi\big(\rho_x(1 - (\tfrac{1}{2} + \tfrac{2\log 2}{\rho_x^2}))/\sqrt{2}\big) - \Phi\big((\tfrac{1}{2} + \tfrac{2\log 2}{\rho_x^2})\rho_x/\sqrt{2}\big) + \tfrac{1}{2} \\
&= \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) - \tfrac{1}{2}\Phi\big(\rho_x(1 - (\tfrac{1}{2} + \tfrac{2\log 2}{\rho_x^2}))/\sqrt{2}\big) - \Phi\big((\tfrac{1}{2} + \tfrac{2\log 2}{\rho_x^2})\rho_x/\sqrt{2}\big) + 1 \\
&= \tfrac{3}{2}\Phi(\rho_x/2)^2 - \Phi(\rho_x/2) - \tfrac{1}{2}\Phi\big(\tfrac{\rho_x}{2} - \tfrac{2\log 2}{\rho_x})/\sqrt{2}\big) - \Phi\big((\tfrac{\rho_x}{2} + \tfrac{2\log 2}{\rho_x})/\sqrt{2}\big) + 1 \\
&= \underbrace{\tfrac{3}{2}(\Phi(\rho_x/2) - \tfrac{1}{3})^2}_{(t_1)} - \underbrace{\tfrac{1}{2}\Phi\big(\tfrac{\rho_x}{2} - \tfrac{2\log 2}{\rho_x})/\sqrt{2}\big)}_{(t_2)} - \underbrace{\Phi\big((\tfrac{\rho_x}{2} + \tfrac{2\log 2}{\rho_x})/\sqrt{2}\big) + \tfrac{5}{6}}_{(t_3)}
\end{aligned}$$

As ρ_x increases, t_1 rises, while t_2 and t_3 fall. For any interval $[a, b]$, a minimizes t_1 while b minimizes t_1 and t_2. Then $\min_{\rho\in[a,b]}\big(t_1(\rho)+t_2(\rho)+t_3(\rho)\big) > \min_{\rho\in[a,b]} t_1(\rho) + \min_{\rho\in[a,b]} t_2(\rho) + \min_{\rho\in[a,b]} t_3(\rho) = t_1(a) + t_2(b) + t_3(b)$.

Then, we can separate ρ_x into two separate values ρ_a and ρ_b to define a strict lower bound for $\Delta(\rho_x)$:

$$\tfrac{3}{2}(\Phi(\underline{\rho_{\mathbf{b}}}/2) - \tfrac{1}{3})^2 - \tfrac{1}{2}\Phi\big(\underline{\rho_{\mathbf{a}}}/2 - 2\log 2/\underline{\rho_{\mathbf{a}}})/\sqrt{2}\big) - \Phi\big((\underline{\rho_{\mathbf{a}}}/2 + 2\log 2/\underline{\rho_{\mathbf{a}}})/\sqrt{2}\big) + \tfrac{5}{6}$$

We can then analytically find an arbitrary covering of intervals over $[2\sqrt{\log 2}, 4.2)$ with positive lower bounds:

Interval Endpoints	Lower Bound on $\Delta(\rho)$
$(2\sqrt{\log 2},\ 1.8]$	0.0059
$[1.8,\ 2.0]$	0.0053
$[2.0,\ 2.2]$	0.0159
$[2.2,\ 2.4]$	0.0256
$[2.4,\ 2.8]$	0.0104
$[2.8,\ 3.4]$	0.0065
$[3.4,\ 4.2]$	0.0094

Aggregating over all these intervals, we have $\min_{\epsilon\in[2\sqrt{\log 2},4.2)} \Delta(\rho) > 0.0053 > 0$, so $T_\Delta\big(\min\{1,\epsilon^*\}\big) > 0$ over $\rho \in (2\sqrt{\log 2}), 4.2)$. Earlier, we showed that $T_\Delta\big(\min\{1,\epsilon^*\}\big) > 0$ over $\rho \in (0, 2\sqrt{\log 2}]$ and $\rho \in (4.2, \infty)$.

Putting this all together, $T_\Delta(\rho)$ is guaranteed to be positive over all $\rho \in (0,\infty)$. Therefore, $T_{\text{disc}} > T_{\text{raw}}$ for all $\rho_x > 0$ and all $\epsilon \in [0, 1)$.