Ask a Strong LLM Judge when Your Reward Model is Uncertain

Zhenghao Xu, Qin Lu, Qingru Zhang, Liang Qiu, Ilgee Hong, Changlong Yu, Wenlin Yao, Yao Liu, Haoming Jiang, Lihong Li, Hyokun Yun, Tuo Zhao

Georgia Tech · amazon SFAI · Neural Information Processing Systems

Background and Motivation

▷ **Motivation:** Reward Models (RM) are central to RLHF but suffer from **reliability** and **efficiency** issues.

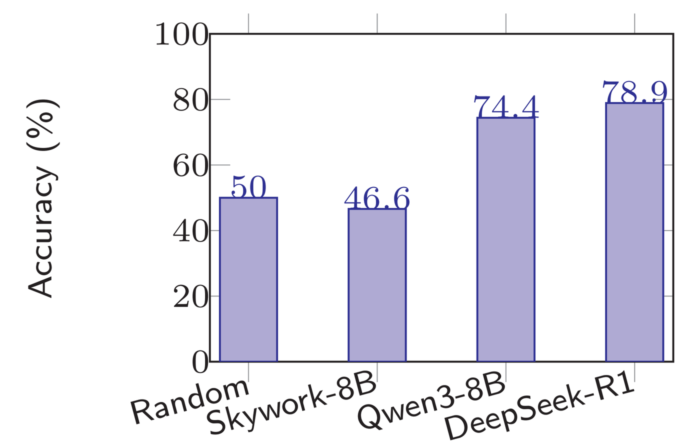
▷ **Bradley-Terry (BT)/classification RM:** Fast but brittle.

- **Output:** Scalar pointwise reward or pairwise preference score
- **Pros:** Low inference cost ($\sim$ generate 1 token)
- **Cons:** Poor OOD generalization
  - Overfit to spurious patterns in limited training data, e.g., style, tense, format biases
  - Require large amount of data to fix/mitigate

▷ **Strong LLM judges w/ reasoning:** More reliable but slow.

- **Output:** Long CoT and final verdict of better response
- **Pros:** Strong performance gain from CoT
  - Can identify key attributes from superficial ones, e.g., correctness from format
- **Cons:** High inference cost (often $>$ 1k tokens) might block the online RLHF training loop

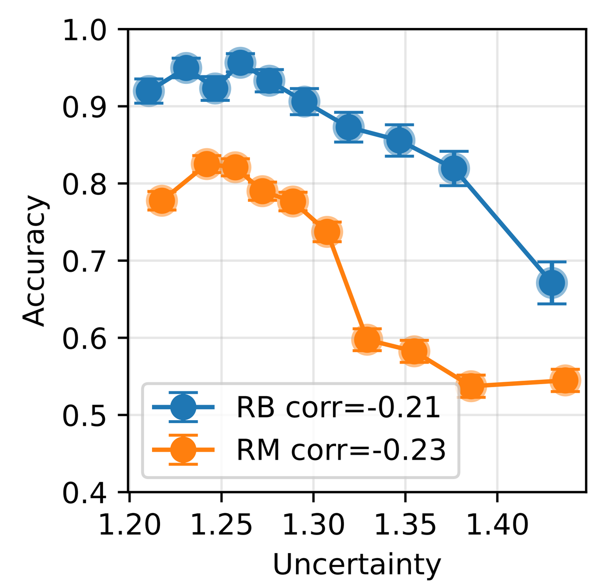
BT RM vs LLM judges on RM-Bench (Hard)



▷ **Question:** *How to get LLM judge quality at low cost?*

▷ **Insight:** Not all queries are equally difficult for RM

- High uncertainty $\to$ Low accuracy $\to$ Routing to strong LLM judge



Pairwise Uncertainty Quantification

▷ **Pointwise BT RM:** prompt x, response $y \to r(x, y)$

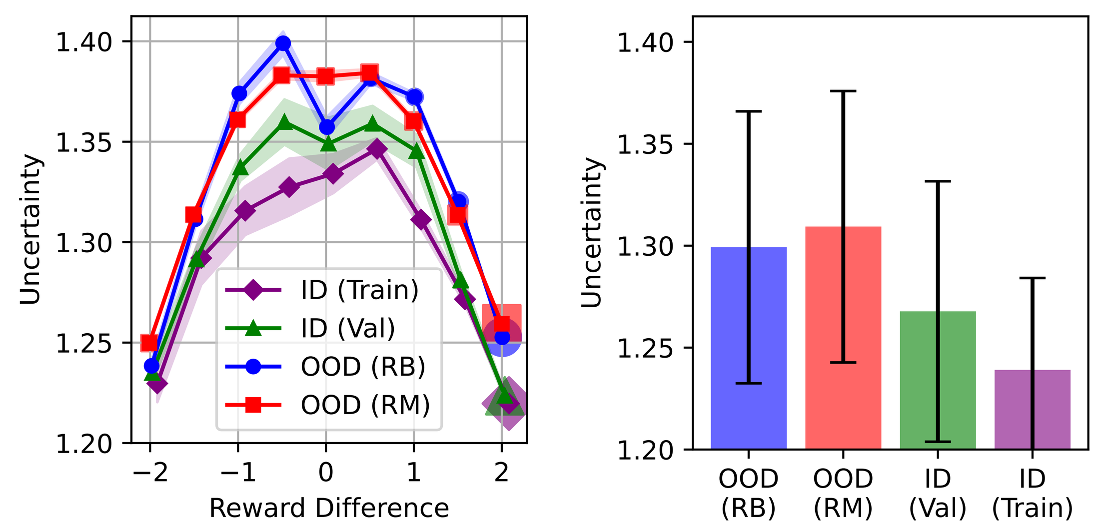
- $\max_r \ \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}}[\log(\sigma(r(x, y_w) - r(x, y_l)))]$
- **Issue:** Solution $r(x, y) \approx r^\star(x, y) + s(x)$ not unique, hard to assign prior, UQ ill-posed

▷ **Pairwise RM:** prompt x, response $y_1, y_2 \to p(x, y_1, y_2)$

- $\max_r \ \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}}[\log(\sigma(p(x, y_w, y_l)))]$
- **Benefit:** Well-defined binary classification, yields generalized preference model, principled UQ possible

▷ **UQ Method:** Spectral-Normalized Gaussian Processes (SNGP, Liu et al. 2023)

- **Architecture:** SN layer and GP head
- **Logit** $g(h)$: Measures preference strength (Aleatoric).
- **Posterior covariance** $u(x, y_1, y_2)$: Measures distance to training data (Epistemic).
- **Prediction:** $p(x, y_1, y_2) = g(h)/u(x, y_1, y_2)$
- **Benefits:** Single model (vs ensemble), single inference (vs MC dropout), distance awareness



Uncertainty-Based Routing

▷ **Routing Strategy:** Define threshold $\bar{u}$, route highly uncertain pairs to the strong LLM judge, get $\widetilde{p}(x, y_i, y_j) \approx r^\star(x, y_i) - r^\star(x, y_j)$

$$\widetilde{p}(x, y_i, y_j) = \begin{cases} p(x, y_i, y_j), & u \le \bar{u} \quad \text{(Cheap)} \\ J(x, y_i, y_j), & u > \bar{u} \quad \text{(Accurate)} \end{cases}$$

Assume LLM judge's verdict is reliable

$$J(\boldsymbol{x}, \boldsymbol{y}_i, \boldsymbol{y}_j) = \begin{cases} \sigma^{-1}(1-\epsilon), & \boldsymbol{y}_i \overset{J}{\succ} \boldsymbol{y}_j \\ \sigma^{-1}(\epsilon), & \boldsymbol{y}_i \overset{J}{\prec} \boldsymbol{y}_j \\ \sigma^{-1}(1/2), & \boldsymbol{y}_i \overset{J}{\sim} \boldsymbol{y}_j \end{cases}$$

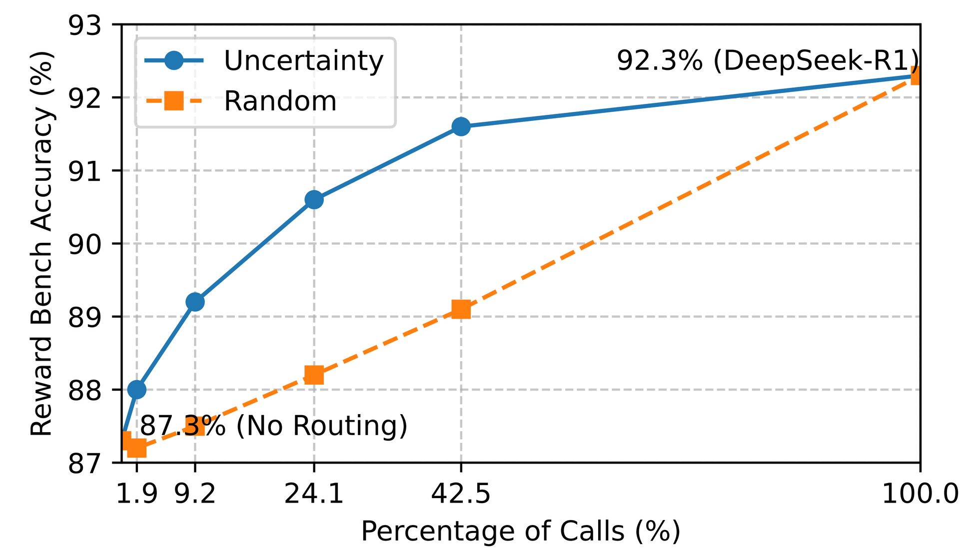
Reward Evaluation Results

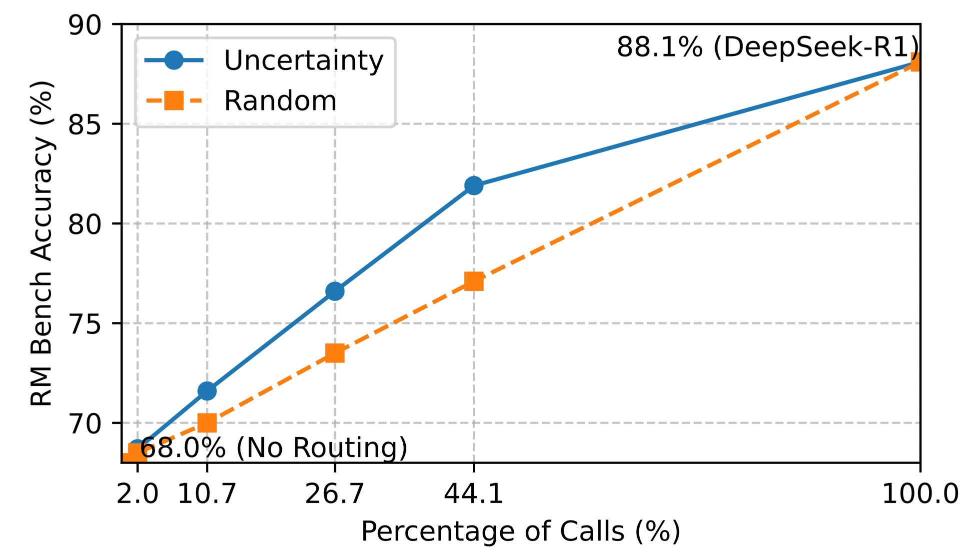
▷ **Improved Accuracy-Cost Trade-off**

- **Baseline:** random routing with the same number of judge calls

Routing	Num of Calls	Reward Bench (%)				
		chat	chat hard	safety	reasoning	avg. (vs rand)
No routing	0	95.8	73.8	89.4	90.0	87.3
Uncertainty	58 (1.9%)	96.1	74.8	89.5	91.7	**88.0 (+0.8)**
	274 (9.2%)	96.4	76.8	89.8	93.7	**89.2 (+1.7)**
	719 (24.1%)	96.9	80.3	89.8	95.4	**90.6 (+2.4)**
	1270 (42.5%)	98.3	81.2	90.0	97.0	**91.6 (+2.5)**
Random	58 (1.9%)	95.5	74.0	89.4	89.9	87.2
	274 (9.2%)	96.4	73.7	89.5	90.4	87.5
	719 (24.1%)	95.0	75.9	90.2	91.5	88.2
	1270 (42.5%)	95.5	77.5	91.6	91.9	89.1
DeepSeek-R1	100%	95.5	85.8	91.1	96.9	92.3

Routing	Num of Calls	RM Bench (%)							
		chat	math	code	safety	easy	normal	hard	avg. (vs rand)
No routing	0	67.1	59.5	54.2	91.2	87.2	72.0	44.9	68.0
Uncertainty	242 (2.0%)	68.7	60.0	54.7	91.4	87.4	73.3	45.5	**68.7 (+0.2)**
	1285 (10.7%)	69.6	64.9	59.7	92.0	89.1	76.3	49.2	**71.6 (+1.6)**
	3188 (26.7%)	71.3	73.8	68.7	92.6	91.4	81.5	56.9	**76.6 (+3.1)**
	5270 (44.1%)	73.2	83.5	78.3	92.7	93.6	86.9	65.3	**81.9 (+4.8)**
Random	242 (2.0%)	67.5	60.2	54.9	91.2	87.4	72.4	45.6	68.5
	1285 (10.7%)	68.0	63.6	57.2	91.3	88.0	74.1	48.0	70.0
	3188 (26.7%)	69.6	69.3	63.6	91.3	89.4	77.0	54.0	73.5
	5270 (44.1%)	70.1	76.8	69.9	91.6	91.0	81.3	59.1	77.1
DeepSeek-R1	100%	76.8	95.7	87.8	92.0	94.0	91.3	78.9	88.1





- **Inference time comparison:** Uncertain pairs take more reasoning time, reflecting their difficulty
  - Still higher accuracy than random in same time

Trigger Threshold	10	1.45	1.4	1.35	1.3	<1
Num of Calls (Ratio)	0%	2.0%	10.7%	26.7%	44.1%	100%
Inference Time (s) - Uncertainty	518	632	1113	2200	3007	5642
Inference Time (s) - Random	518	625	1093	1979	2615	5642

Downstream Alignment

▷ **Policy Gradient from Preferences:** For each prompt x, sample $y_1, \ldots, y_K \sim \pi_\theta(\cdot \mid x)$, minimize the PG loss

$$\mathcal{L}_{\text{policy}}(\theta) = -\frac{1}{K}\sum_{i=1}^{K} A_i \log \pi_\theta(y_i|x_i)$$

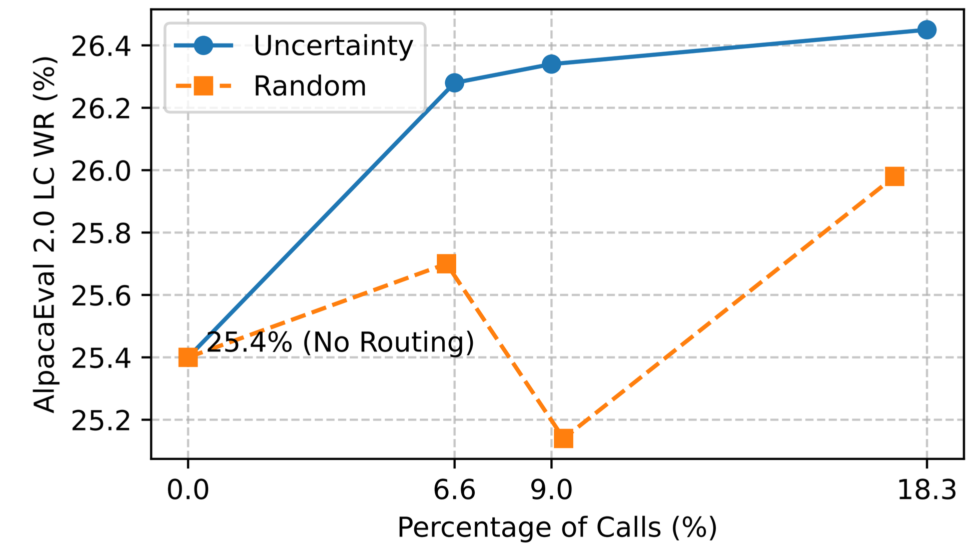
We use RLOO advantage estimator, which can be recovered from pairwise preference $\widetilde{p}$:

$$A_i = r(x, y_i) - \frac{1}{K-1}\sum_{j\neq i} r(x, y_j) = \frac{1}{K-1}\sum_{j\neq i}\underbrace{[r(x, y_i) - r(x, y_j)]}_{\widetilde{p}(x, y_i, y_j)}$$

- Also applies to other PG methods using MC samples and mean baseline, e.g., GRPO

▷ **Downstream Alignment Results**

Model	Num of Calls	Arena-Hard (%)	AlpacaEval 2.0 (%)		MT-Bench		
		v0.1 WR	LC WR	WR	Turn 1	Turn 2	Avg
Base model	-	24.5	22.31	23.63	7.98	6.80	7.47
No routing (10.0)	0	28.1	25.40	27.35	**8.19**	6.98	7.65
Uncertainty (1.35)	7668 (6.6%)	28.9	26.28	**28.97**	8.05	7.19	7.65
Uncertainty (1.30)	10522 (9.0%)	28.9	26.34	28.53	8.03	7.13	7.63
Uncertainty (1.20)	21363 (18.3%)	**29.8**	**26.45**	28.91	7.95	**7.40**	**7.71**
Random (1.35)	7523 (6.4%)	26.5	25.70	28.55	8.09	7.20	**7.71**
Random (1.30)	10854 (9.3%)	27.7	25.14	28.29	7.93	6.74	7.41
Random (1.20)	20474 (17.5%)	28.5	25.98	28.51	8.00	6.62	7.45



Takeaways

We propose an **Uncertainty-Based Routing Framework**

- **Efficiency:** Call costly judge only for uncertain samples
- **Effectiveness:** SNGP identifies OOD samples
- **Impact:** More reliable online RLHF with less computational overhead
- **Future directions:**
  - Active RM learning, filter samples for further annotations
  - Combine with specialized reasoning RMs