Carnegie Mellon University

C3.ai

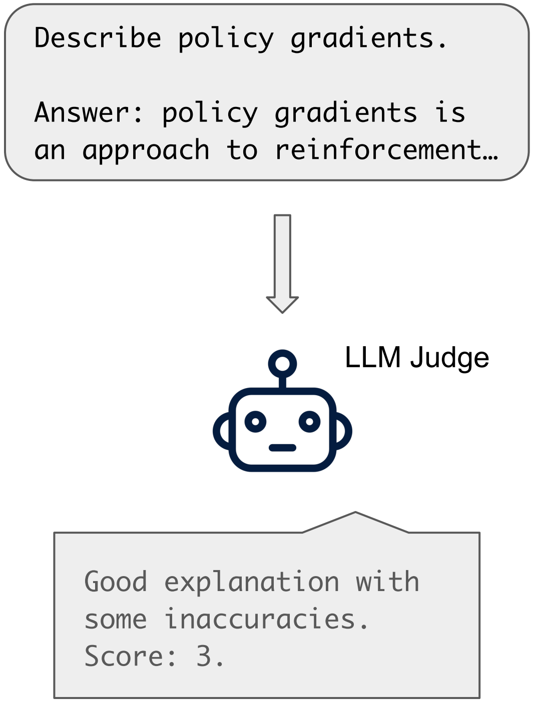
Better Instruction-Following with Minimum Bayes Risk

ICLR 2025 Spotlight

Ian Wu, Patrick Fernandes, Amanda Bertsch, Seungone Kim, Sina Pakazad, Graham Neubig

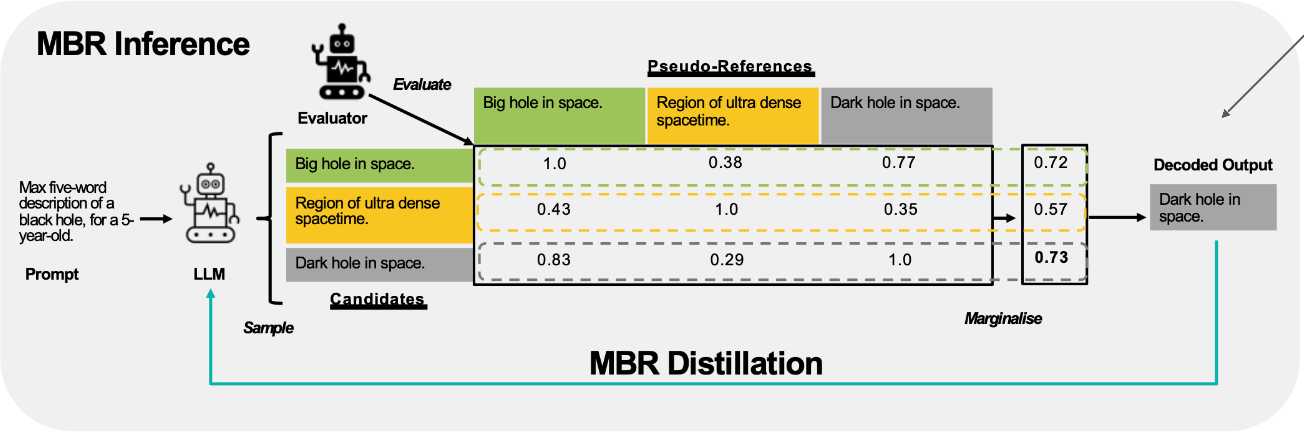
LLM Judges as Supervisors

- **LLMs judges** are widely used for evaluating the quality of text.
- LLM judges may be few-shot prompted LLMs or specialist models trained for judging.
- Instead of using judges for evaluation, it is also possible to use them for **supervision**. This is typically done using **Best-of-*N* decoding**.



Minimum Bayes Risk Decoding

Highest **consensus quality** output



Best-of-*N* Decoding (Reference-Free)

$$\hat{y} = \arg\max_{y \in \mathcal{H}_{\text{hyp}}} u(y).$$

MBR Decoding (Reference-Based)

$$\hat{y} = \arg\max_{y \in \mathcal{H}_{\text{hyp}}} \underbrace{\mathbb{E}_{y^* \sim p(y|x)}[u(y, y^*)]}_{\approx \frac{1}{N_{\text{cand}}} \sum_{j=1}^{N_{\text{cand}}} u(y, y^{(j)})},$$

Experiment I: MBR Inference

	2-7B	**2-13B**	**2-70B**	**3-8B**	**3-70B**	**Avg. Δ**
Greedy	14.4	19.0	22.8	34.4	42.7	0
BS	14.8	18.2	21.5	33.9	42.4	-0.50
Longest	10.5	15.2	19.8	29.8	40.4	-3.51
Prometheus BoN	16.4	20.8	25.0	35.5	44.3	1.74
ROUGE MBR	16.2	20.0	24.7	35.4	43.7	1.33
BERTScore MBR	16.2	20.5	24.4	35.7	44.0	1.50
SFR-Embedder MBR	12.1	16.6	22.2	32.5	42.8	-1.42
Prometheus MBR	**17.7**	**23.4**	**26.2**	**37.9**	**46.0**	**3.62**

Table 1: AlpacaEval 2.0 win rates (%) for various models and decoding strategies, along with the average win rate differences compared to greedy decoding across all models (denoted as **Avg.** Δ). MBR decoding with Prometheus consistently outperforms all baseline methods and other MBR decoding methods.

Key Takeaway

MBR inference with LLM judge Prometheus 2 improves performance on AlpacaEval.

Experiment I: MBR Inference

	2-7B	**2-13B**	**2-70B**	**3-8B**	**3-70B**	**Avg. Δ**
Greedy	5.72	5.90	6.50	7.54	8.29	0
BS	5.58	5.95	6.49	7.30	8.20	-0.09
Longest	5.67	6.03	6.59	7.22	8.22	-0.04
Prometheus BoN	5.77	6.08	6.65	7.66	8.42	0.13
ROUGE MBR	5.78	6.11	6.68	7.63	8.31	0.11
BERTScore MBR	5.68	6.02	6.72	7.52	8.42	0.08
SFR-Embedder MBR	5.73	6.04	6.54	7.45	8.33	0.03
Prometheus MBR	**6.10**	**6.26**	**6.79**	**7.69**	**8.50**	**0.28**

Table 2: MT-Bench scores for various models and decoding strategies, along with the average score differences compared to greedy decoding across all models (denoted as **Avg. Δ**). MBR decoding with Prometheus consistently outperforms all baseline methods and other MBR decoding methods.

Key Takeaway

MBR inference with LLM judge Prometheus 2 improves performance on MT-Bench.

Experiment I: MBR Inference

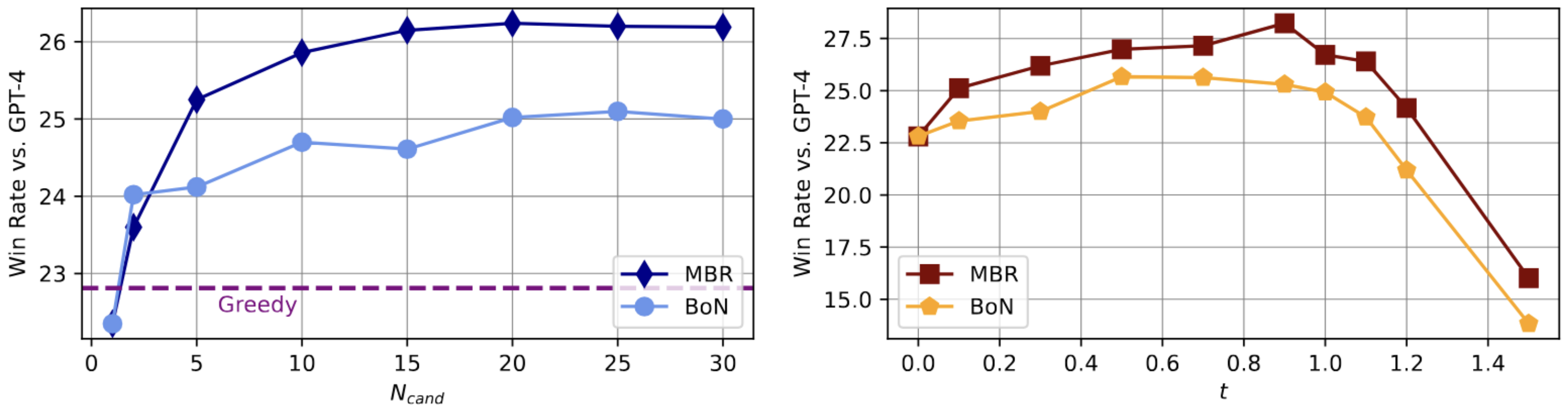


Figure 2: AlpacaEval 2.0 win rates (%) for Llama2-70b with varying hypothesis set size N_{cand} **(left)** and generation temperature t **(right)** values for Prometheus MBR and BoN decoding. Performance for both methods initially increases with N_{cand} and plateaus at around $N_{\text{cand}} = 20$. Performance also initially increases with t, but drops rapidly after $t = 1.0$.

Experiment I: MBR Inference

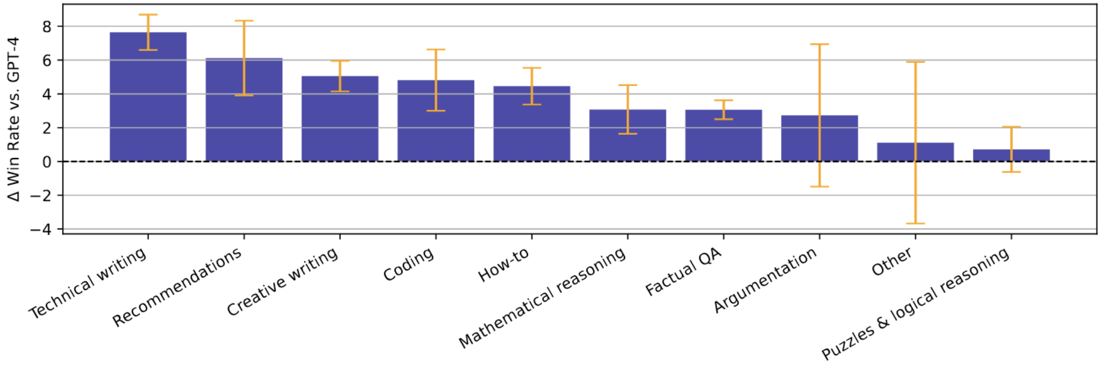


Figure 3: Difference in AlpacaEval 2.0 win rates (%) between Prometheus MBR decoding and greedy decoding averaged over all five LLMs and broken down by question category. A positive value indicates that MBR decoding outperforms greedy decoding on the given category. Orange bars represent the standard error. We find that Prometheus MBR decoding improves performance across a wide range of question categories.

Key Takeaway

MBR inference improves performance across a range of tasks. Improvements are most significant for writing-based tasks.

Experiment I: MBR Inference

	2-7B	2-13B	2-70B	3-8B	3-70B	Avg. Δ
Greedy	5.72	5.90	6.50	7.54	8.29	0
Prometheus-2-7B-BoN	5.77	6.08	6.65	7.66	8.42	0.13
Prometheus-2-7B-MBR	6.10	6.26	6.79	7.69	8.50	0.28
Prometheus-2-8x7B-BoN	6.01	6.17	6.80	7.75	8.41	0.24
Prometheus-2-8x7B-MBR	**6.26**	6.32	6.87	7.79	**8.64**	0.39
JudgeLM-7b-BoN	5.63	5.95	6.69	7.37	8.26	-0.01
JudgeLM-7b-MBR	6.00	6.11	6.79	7.69	8.44	0.22
JudgeLM-33b-BoN	5.68	6.03	6.58	7.37	8.35	0.01
JudgeLM-33b-MBR	5.94	6.27	6.88	**7.92**	8.50	0.31
Llama3-8b-Instruct-BoN	5.83	6.05	6.61	7.60	8.38	0.10
Llama3-8b-Instruct-MBR	5.96	6.28	6.84	7.80	8.47	0.28
Llama3-70b-Instruct-BoN	5.77	6.16	6.57	7.39	8.35	0.06
Llama3-70b-Instruct-MBR	6.22	**6.43**	**6.94**	7.87	8.52	**0.41**

Table 3: MT-Bench scores for BoN and MBR decoding with various judge LLMs as utility metrics, along with the average score differences compared to greedy decoding across all models (denoted **Avg. Δ**). MBR decoding consistently outperforms BoN decoding across all comparable utility metrics.

Key Takeaway

MBR inference improvements generalise across various judges and judge scales.

Experiment II: MBR Distillation

- Train the generator LLM on its own best and worst outputs, as determined by MBR, via **DPO**.

- Enables training without human-generated labels or preferences!

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x, \hat{y}^+, \hat{y}^- \sim \mathcal{Y}_k)} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(\hat{y}^+ | x)}{\pi_{\text{ref}}(\hat{y}^+ | x)} - \beta \log \frac{\pi_\theta(\hat{y}^- | x)}{\pi_{\text{ref}}(\hat{y}^- | x)} \right) \right]$$

$$\hat{y}^+ = \underset{y \in \mathcal{H}_{\text{hyp}}}{\arg\max}\, \tilde{u}(y) \qquad \hat{y}^- = \underset{y \in \mathcal{H}_{\text{hyp}}}{\arg\min}\, \tilde{u}(y)$$

Experiment II: MBR Distillation

	AlpacaEval 2.0		MT-Bench	
	7B	**13B**	**7B**	**13B**
sft w. Greedy	5.18	8.24	5.43	5.85
sft w. MBR	**9.99**	13.6	5.78	6.31
sft-full	6.35	9.40	5.55	6.26
dpo-1-BoN	5.78	10.3	5.78	6.08
dpo-2-BoN	6.22	11.2	5.91	6.41
dpo-3-BoN	6.40	12.8	5.88	6.56
dpo-1-MBR	5.68	10.8	5.78	6.48
dpo-2-MBR	7.22	13.9	6.11	6.73
dpo-3-MBR	8.86	**15.3**	**6.14**	**6.75**

	AlpacaEval 2.0	MT-Bench
sft-1-MBR	5.52	5.48
sft-2-MBR	6.75	5.43
sft-3-MBR	6.48	5.51

Table 4: **(Left)** AlpacaEval 2.0 win rates (%) and MT-Bench scores for models self-trained using DPO. After three rounds of training, the self-trained models consistently outperform their BoN counterparts and SFT baselines. **(Top)** AlpacaEval 2.0 win rates (%) and MT-Bench scores for models self-trained using SFT. Self-training with SFT yields substantially worse results than self-training with DPO.

Key Takeaway

MBR distillation with DPO improves greedy decoding performance.

Conclusion

- MBR decoding yields significant and consistent improvements to model performance relative to Best-of-N decoding.
- MBR decoding can be used to curate self-training data to further improve greedy performance.
- We hope that our work inspires future work on MBR decoding as well as usage of LLM judges for supervision.