Temper and Tilt Lead to SLOP: Reward Hacking Mitigation with Inference-Time Alignment

Ye Wang [1] **Jing Liu** [1] **Toshiaki Koike-Akino** [1]

Abstract

Inference-time alignment techniques offer a lightweight alternative or complement to costly reinforcement learning, while enabling continual adaptation as alignment objectives and reward targets evolve. Existing theoretical analyses justify these methods as approximations to sampling from distributions optimally tilted toward a given reward model. We extend these techniques by introducing reference-model temperature adjustment, which leads to further generalization of inference-time alignment to ensembles of generative and reward models combined as a sharpened logarithmic opinion pool (SLOP). To address reward hacking, we propose an algorithm for calibrating SLOP weight parameters and experimentally demonstrate that it improves robustness while preserving alignment performance.

1. Introduction

Reinforcement learning (RL) is widely used to realize effective and aligned foundation models (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Korbak et al., 2022a; Bai et al., 2022; Rafailov et al., 2023; Shao et al., 2024). Inference-time alignment methods, such as Best-of-N (BoN) (Stiennon et al., 2020; Nakano et al., 2021), provide a lightweight alternative or complement to RL, have been shown to perform competitively (Gao et al., 2023; Dubois et al., 2023), and enable flexible continual adaptation to evolving alignment objectives.

Common across many RL approaches is the optimization of the policy π to maximize expected reward, subject to KL-regularization with respect to a reference policy p, i.e., $\max_\pi \mathbb{E}_\pi[r] - \lambda \mathrm{KL}(\pi \| p)$. Korbak et al. (2022b) observed that, in principle, the optimal policy $\pi^* \propto p \exp(r/\lambda)$, which led to a line of research (Yang et al., 2024b; Mroueh & Nitsure, 2025; Verdun et al., 2025; Khalaf et al., 2025) that analyzed how well BoN and related inference-time methods approximate sampling this optimal policy.

Our work extends upon these inference-time alignment methods, by introducing reference model temperature adjustment, which is closely related to a proposal of Jinnai et al. (2025), and we further generalize to ensembles of generative and reward models as a form of logarithmic opinion pool (Heskes, 1997). We experimentally demonstrate the effectiveness of calibrating the ensemble weights to mitigate reward hacking, while preserving alignment performance, for visual question answering and math reasoning tasks.

2. Formulation and Methodology

Tempered and Tilted Alignment. Let $p(y \mid x)$ denote a reference model and let $r(x, y)$ denote a proxy reward model. We consider the following tempered variant of KL-regularized reward maximization:

$$\pi^*_{\alpha,\beta}(\cdot \mid x) = \arg\max_\pi \mathbb{E}\big[R_{\alpha,\beta}(x, y)\big] - \mathrm{KL}_x(\pi \| p), \quad (1)$$

where the expectation is over $y \sim \pi(\cdot \mid x)$, $R_{\alpha,\beta}(x, y) := \beta r(x, y) + (\alpha - 1) \log p(y \mid x)$, with $\alpha, \beta \in \mathbb{R}$, and $\mathrm{KL}_x(\pi \| p) := \mathrm{KL}\big(\pi(\cdot \mid x) \| p(\cdot \mid x)\big)$. The standard KL-regularized objective is recovered when $\alpha = 1$, in which case the solution is the usual reward-tilted distribution (Korbak et al., 2022b). More generally, (1) has the solution

$$\pi^*_{\alpha,\beta}(y \mid x) = \frac{p(y \mid x)^\alpha \exp(\beta r(x, y))}{C_{\alpha,\beta,x}}, \quad (2)$$

where $C_{\alpha,\beta,x}$ is a normalizing constant. Thus, α acts as an inverse temperature on the reference model, while β controls the strength and direction of the reward tilt. In particular, $\beta > 0$ favors high-reward outputs, $\beta = 0$ reduces to sampling from the tempered reference model, and $\beta < 0$ allows the proxy reward to be inverted when it is misaligned.

Sharpened Logarithmic Opinion Pooling. Equation (2) can be viewed as a logarithmic opinion pool (Heskes, 1997) between the reference model and the reward-induced align-

[1]Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. Correspondence to: Ye Wang <yewang@merl.com>.

Published at the Second Workshop on Agents in the Wild: Safety, Security, and Beyond (AIWILD) at ICML 2026. Copyright 2026 by the author(s).

ment distribution $q_r(y \mid x) \propto \exp(r(x,y))$:

$$\pi^*_{\alpha,\beta}(y \mid x) \propto p(y \mid x)^{\alpha} q_r(y \mid x)^{\beta} \quad (3)$$

$$\propto \exp\left(\alpha \log p(y \mid x) + \beta r(x,y)\right). \quad (4)$$

This perspective naturally generalizes to an ensemble of generative and reward models. Let $s_i(x,y)$ denote the score contributed by expert i, where $s_i = \log p_i$ for a generative model and $s_i = r_i$ for a reward model. For weights $\omega \in \mathbb{R}^m$, we define the sharpened logarithmic opinion pool (SLOP),

$$\pi^*_{\omega}(y \mid x) = \frac{\exp\left(\sum_{i=1}^{m} \omega_i s_i(x,y)\right)}{C_{\omega,x}}. \quad (5)$$

Unlike classical logarithmic opinion pools, we do not constrain the weights to be nonnegative or to sum to one. This allows SLOP to sharpen reliable experts, suppress weak ones, and even invert anti-aligned reward models through negative weights.

Inference-Time Approximation. Directly sampling from (5) is generally intractable over long sequences. We approximate SLOP with the following extension of Soft Best-of-N (SBoN) (Verdun et al., 2025): given a prompt x, sample n candidates $y_1, \ldots, y_n \overset{\text{iid}}{\sim} p_1(\cdot \mid x)$, evaluate expert scores $s_i(x, y_j)$, and select one candidate according to

$$\widehat{\pi}_{\omega,n}(y_j \mid x) \propto \exp\left(\sum_{i=1}^{m} \omega_i s_i(x,y_j) - s_1(x,y_j)\right), \quad (6)$$

where $s_1(x,y) = \log p_1(y \mid x)$ is subtracted to effectively adjust the first expert weight to $(\omega_1 - 1)$, which accounts for the candidates being already sampled from the first expert. In the special case of one reference model and one reward model, this approximates tempered-and-tilted sampling with likelihoods proportional to $\exp\big((\alpha-1)\log p(y_j \mid x) + \beta r(x,y_j)\big)$, where (α,β) correspond to (ω_1,ω_2).

Calibrating SLOP Weights to Mitigate Reward Hacking. Proxy rewards may be misaligned with the desired objective, so increasing their weights may amplify reward hacking. We calibrate ω using a small set of prompts with access to the gold or verifiable reward g. For prompts $x_1, \ldots, x_k$, we sample candidates from p_1, compute expert scores and gold rewards, and choose ω to maximize the empirical gold reward induced by the candidate-level SLOP distribution:

$$\widehat{J}(\omega) = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{n} \widehat{\pi}_{\omega,n}(y_{ij} \mid x_i) g(x_i, y_{ij}) - \lambda\|\omega\|^2, \quad (7)$$

where $\lambda \geq 0$ is optionally used to regularize the weights and avoid fully sharpening to hard outputs. After calibration, the learned weights are fixed and used for inference-time SLOP sampling without further access to g. Thus, calibration provides a lightweight continual-adaptation mechanism: when reward targets or failure modes shift, only the inference-time pooling weights need to be updated.

Algorithm 1 Calibrating SLOP Weights

Require: reference model p_1; expert models $s_1, \ldots, s_m$, where $s_1(x,y) = \log p_1(y \mid x)$; calibration prompts $x_1, \ldots, x_k$; candidates per prompt n; gold reward g; steps T; learning rate η; weight decay parameter λ;
  for $i = 1$ to k **do**
    for $j = 1$ to n **do**
      Sample candidates $y_{ij} \sim p_1(\cdot \mid x_i)$
      Evaluate gold rewards $g_{ij} \leftarrow g(x_i, y_{ij})$
      Score samples $s_{\ell ij} \leftarrow s_\ell(x_i, y_{ij})$, for $\ell = 1, \ldots, m$
    end for
  end for
  Initialize $\omega^{(0)} \leftarrow (1, \ldots, 1)$
  for $t = 0$ to $T-1$ **do**
    for $i = 1$ to k **do**
      Compute candidate logits

$$a_{ij} \leftarrow \sum_{\ell=1}^{m} \omega_\ell^{(t)} s_{\ell ij} - s_{1ij}, \quad \text{for } j = 1, \ldots, n$$

      Compute candidate weights

$$\pi_{ij} \leftarrow \frac{\exp(a_{ij})}{\sum_{q=1}^{n}\exp(a_{iq})}, \quad \text{for } j = 1, \ldots, n$$

    end for
    Compute objective estimate

$$\widehat{J}(\omega^{(t)}) \leftarrow \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{n} \pi_{ij} g_{ij} - \lambda\|\omega^{(t)}\|^2$$

    Update $\omega^{(t+1)} \leftarrow \omega^{(t)} + \eta \nabla_\omega \widehat{J}(\omega^{(t)})$
  end for
  return $\omega^{(T)}$

3. Experimental Results and Discussion

3.1. Reward-Guided Visual Question Answering

This experiment considers the task of multiple-choice, visual question answering, evaluated with the ScienceQA (SQA) (Lu et al., 2022) benchmark dataset. The gold reward is one for the correct answer and zero otherwise. Thus, the objective is to maximize accuracy. To investigate the impact of imprecise reward models, we take a pretrained VLM as the generative reference model and pair it with a synthetic proxy reward model that may instead reward incorrect answers, at some given error rate. Since SQA multiple-choice questions contain only two to five answer options, we constrain VLM generation to single-token decoding over the small response set $\mathcal{Y} = \{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\}$. This simplification allows for exact sampling of the tempered-and-tilted (two-expert SLOP) distribution and focuses this controlled experiment on weight optimization for inaccurate proxy reward models. The (α, β) weights are optimized with 200

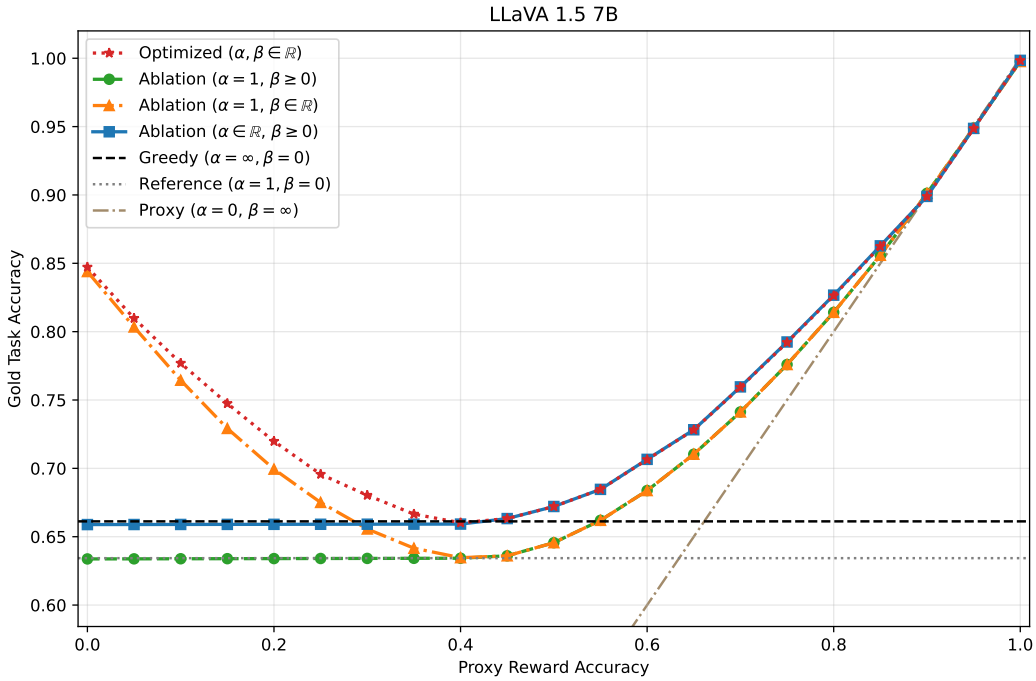

Figure 1. SLOP with LLaVA-1.5-7B paired with synthetic proxy reward with varying accuracy, evaluated on SQA.

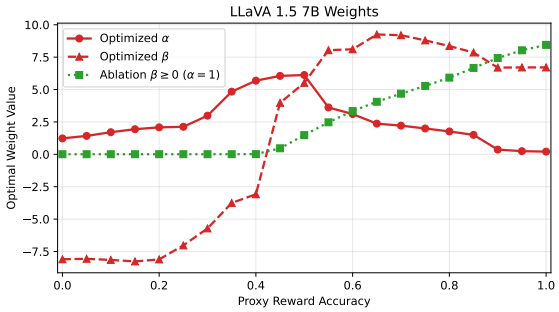

Figure 2. Optimized (two-expert) SLOP weights for SQA.

calibration samples from the SQA test split, while the remaining 4041 samples of the SQA test split are held-out for evaluation. See Appendix B for further details and additional results with other VLMs, which yield similar findings.

Figure 1 plots the results for the LLaVA-1.5-7B (Liu et al., 2023; 2024) VLM paired with a synthetic proxy reward model at varying accuracies. The "optimized" curve is the result of freely optimizing $\alpha, \beta \in \mathbb{R}$, while the "ablation" curves constrain with $\alpha = 1$ and/or $\beta \geq 0$. Two baselines ignore the proxy reward model ($\beta = 0$): sampling the "reference" model ($\alpha = 1$, 63.4% accuracy) and "greedy" token selection ($\alpha = \infty$, 66.1% accuracy). On the other hand, ignoring the reference model and purely following the "proxy" reward ($\alpha = 0$, $\beta = \infty$) yields the proxy reward accuracy, which works well for accurate proxies, but quickly degrades (along the slope-one line, falling off the plot) for inaccurate proxies, which depicts the effect of hacking with a faulty reward model.

The SLOP curves generally dominate the performance of the individual reference and proxy models, by effectively combining the two. For the ablations with $\beta \geq 0$, performance gracefully degrades back to the baselines as the proxy reward model becomes more unreliable. The optimized and ablation curves, with unconstrained β, yield increased performance for low proxy reward accuracies, since below 40% accuracy (which is roughly the random guessing accuracy for SQA), the inaccurate proxy more reliably indicates an incorrect answer, providing a useful signal that can be exploited with weight $\beta < 0$. This can be seen in Figure 2, which plots the optimized (α, β) weights across varying proxy reward accuracies, showing how the confidence weight for each model correspondingly varies. At high proxy accuracies, the optimal α is close to zero, as following the proxy suffices. However, at middling proxy accuracies, the value of α is larger, showing more reliance on the confidence of the reference model. For comparison, it also plots the optimal β for the ablation (with fixed $\alpha = 1$ and $\beta \geq 0$), which simply falls back to 0 at 40% proxy reward accuracy or below. We note that the ablation with fixed $\alpha = 1$ and $\beta \geq 0$ is essentially the HedgeTune method proposed by Khalaf et al. (2025).

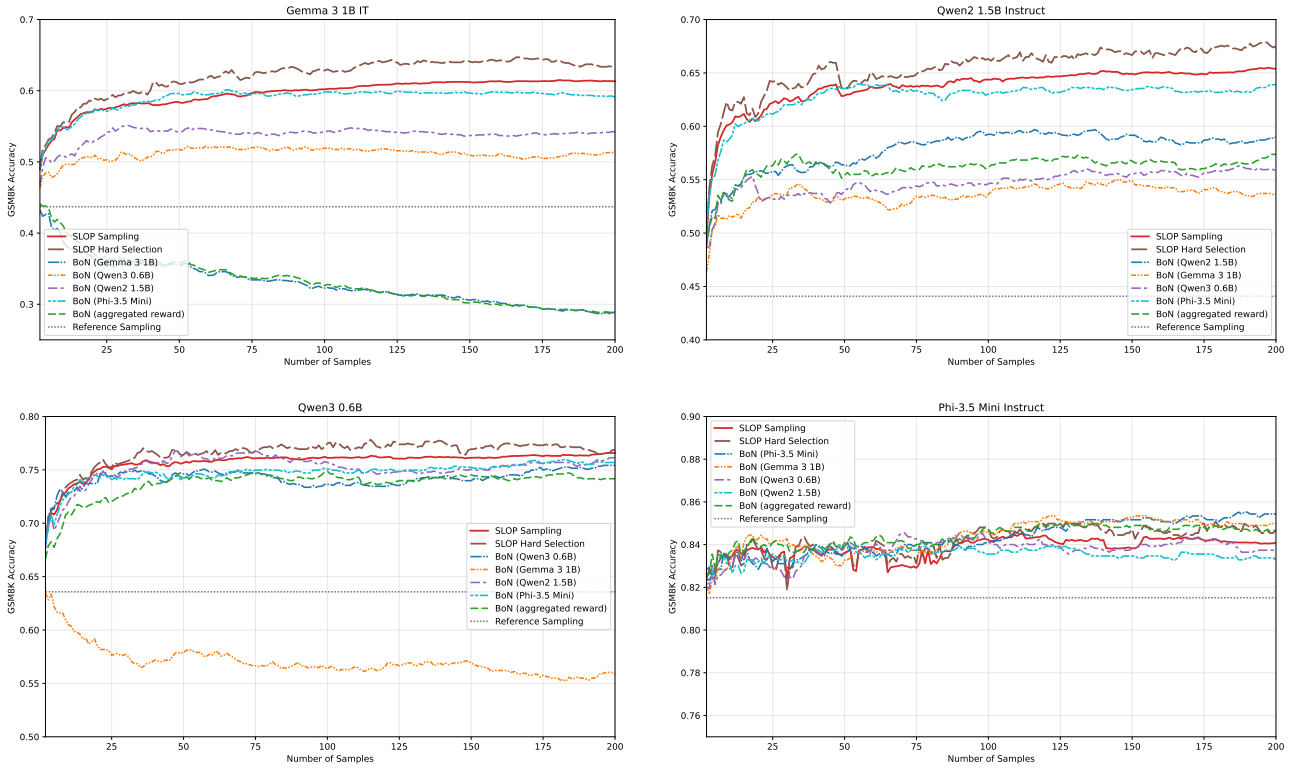

Figure 3. SLOP (with 4 LLMs) evaluated on GSM8K, with different LLMs as the reference model (indicated by the subfigure titles).

3.2. Math Reasoning with Generative Model Pooling

This experiment explores the task of math reasoning, evaluated on the GSM8K (Cobbe et al., 2021) benchmark, with a SLOP that ensembles four LLMs: Gemma-3-1B (Gemma Team et al., 2025), Qwen2-1.5B (Yang et al., 2024a), Qwen3-0.6B (Yang et al., 2025), and Phi-3.5-mini (Abdin et al., 2024). In this task, the input prompt is a grade school math word problem, and responses with the correct numerical solution receive a gold reward of one, and zero otherwise. Thus, the objective is to maximize accuracy.

We apply the approximate SLOP sampling method of (5), with one of the LLMs serving as the reference model to sample up to 200 responses, and the expert scores given by the token-averaged log-likelihoods of each LLM, i.e., $s_i(x, y) = (1/L_y) \log p_i(y \mid x)$, where L_y is the number of tokens in the response y and is introduced to avoid length bias. We apply Algorithm 1 to calibrate the SLOP weights using 200 calibration examples from the GSM8K test split, with the remaining 1119 samples held-out for evaluation.

Figure 3 plots performance, while varying the number of candidate samples n, for each LLM selected as the reference model to generate candidates. The BoN baselines select from the candidates using the individual LLM expert scores or the simple sum of the expert scores as the proxy reward. SLOP sampling tends to outperform the baselines, with more stable, monotonic improvement as the number of samples increases. We also evaluate "hard selection" of the candidate with the highest weighted SLOP score, which further improves performance over SLOP sampling. An exceptional case occurs when the strongest LLM (Phi-3.5) is the reference model, where all methods yield only slight improvement and fall within a few percentage points of each other. Table 1 lists the optimized SLOP weights and baseline model performance. See Appendix C for further details.

Table 1. Reference model baseline sampling accuracy on GSM8K and corresponding optimized SLOP weights.

Model (Acc %)	$\omega_{\text{Gemma-3}}$	ω_{Qwen2}	ω_{Qwen3}	$\omega_{\Phi\text{-3.5}}$
Gemma-3 (43.7)	-0.038	2.55	10.2	33.9
Qwen2 (44.1)	-9.44	-2.36	1.94	35.0
Qwen3 (63.6)	-3.55	12.7	17.0	17.1
Phi-3.5 (81.5)	4.65	6.81	6.12	8.82

3.3. Concluding Remarks and Future Directions

We provide a generalization of inference-time alignment methods that flexibly ensembles multiple generative and/or reward models. Future work will broaden experiments beyond this proof-of-concept, and investigate the impact of correlation or diversity within the ensemble.

Impact Statement

We develop inference-time alignment methods intended to improve robustness to reward hacking and to better combine signals from multiple generative or reward models. Potential positive impacts include more effective deployment of foundational models, especially when proxy rewards are imperfect. Potential negative impacts include misuse of inference-time methods to make models better at satisfying flawed, biased or harmful objectives, or amplification of undesirable model behaviors when the calibration signal is misaligned. Misuse risks are challenging to fully mitigate; however, we emphasize the importance of calibration with gold or verifiable rewards.

References

Abdin, M., Aneja, J., Awadalla, H., Awadallah, A., Awan, A. A., Bach, N., Bahree, A., Bakhtiari, A., Bao, J., Behl, H., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Cai, Q., Chaudhary, V., Chen, D., Chen, D., Chen, W., Chen, Y.-C., Chen, Y.-L., Cheng, H., Chopra, P., Dai, X., Dixon, M., Eldan, R., Fragoso, V., Gao, J., Gao, M., Gao, M., Garg, A., Giorno, A. D., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Hu, W., Huynh, J., Iter, D., Jacobs, S. A., Javaheripi, M., Jin, X., Karampatziakis, N., Kauffmann, P., Khademi, M., Kim, D., Kim, Y. J., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Li, Y., Liang, C., Liden, L., Lin, X., Lin, Z., Liu, C., Liu, L., Liu, M., Liu, W., Liu, X., Luo, C., Madan, P., Mahmoudzadeh, A., Majercak, D., Mazzola, M., Mendes, C. C. T., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Ren, L., de Rosa, G., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Shen, Y., Shukla, S., Song, X., Tanaka, M., Tupini, A., Vaddamanu, P., Wang, C., Wang, G., Wang, L., Wang, S., Wang, X., Wang, Y., Ward, R., Wen, W., Witte, P., Wu, H., Wu, X., Wyatt, M., Xiao, B., Xu, C., Xu, J., Xu, W., Xue, J., Yadav, S., Yang, F., Yang, J., Yang, Y., Yang, Z., Yu, D., Yuan, L., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Aminian, G., Shenfeld, I., Asadi, A. R., Beirami, A., and Mroueh, Y. Best-of-n through the smoothing lens: KL divergence and regret analysis. *arXiv preprint arXiv:2507.05913*, 2025.

Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., and Zhu, K. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pp. 146–158, 1975.

Deng, H. and Raffel, C. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 11781–11791, 2023.

Dubois, Y., Li, C. X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P. S., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Gemma Team, Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Genest, C. and Zidek, J. V. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Heskes, T. Selecting weighting factors in logarithmic opinion pools. *Advances in neural information processing systems*, 10, 1997.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Huang, A., Block, A., Liu, Q., Jiang, N., Krishnamurthy, A., and Foster, D. J. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. In *International Conference on Machine Learning*, pp. 25075–25126. PMLR, 2025.

Jinnai, Y., Morimura, T., Ariu, K., and Abe, K. Regularized best-of-n sampling with minimum bayes risk objective for language model alignment. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 9321–9347, 2025.

Khalaf, H., Verdun, C. M., Oesterling, A., Lakkaraju, H., and Calmon, F. Inference-time reward hacking in large language models. In *Advances in Neural Information Processing Systems*, 2025.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Korbak, T., Elsahar, H., Kruszewski, G., and Dymetman, M. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022a.

Korbak, T., Perez, E., and Buckley, C. RL with KL penalties is better viewed as bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022b.

Kullback, S. and Khairat, M. A note on minimum discrimination information. *The Annals of Mathematical Statistics*, 37(1):279–280, 1966.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *NeurIPS*, 2023.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 26296–26306, 2024.

Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.-W., Zhu, S.-C., Tafjord, O., Clark, P., and Kalyan, A. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Mroueh, Y. and Nitsure, A. Information theoretic guarantees for policy alignment in large language models. *Transactions On Machine Learning Research*, 2025.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Pan, A., Bhatia, K., and Steinhardt, J. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741, 2023.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Verdun, C. M., Oesterling, A., Lakkaraju, H., and Calmon, F. P. Soft best-of-n sampling for model alignment. In *2025 IEEE International Symposium on Information Theory (ISIT)*, pp. 1–6. IEEE, 2025.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Yang, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Liu, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Guo, Z., and Fan, Z. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang,

T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yang, J. Q., Salamatian, S., Sun, Z., Suresh, A. T., and Beirami, A. Asymptotics of language model alignment. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pp. 2027–2032. IEEE, 2024b.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Detailed Exposition of Formulation and Methodology

A.1. Tempered Reward Maximization

We consider a generalization of the KL-regularized reward maximization problem, given by

$$\pi^*_{\alpha,\beta}(\cdot \mid x) = \arg\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\big[\beta r(x,y) + (\alpha-1)\log p(y \mid x)\big] - \mathrm{KL}\big(\pi(\cdot \mid x)\|p(\cdot \mid x)\big), \tag{8}$$

where $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a given (proxy) reward model, $p(y \mid x)$ is a given reference model, and parameters $\alpha, \beta \in \mathbb{R}$ control the regularization. Typically, $\beta > 0$ is used, but we allow for $\beta \leq 0$ to consider situations where the proxy reward model may be severely misaligned with the ideal reward.

The standard KL-regularized reward maximization problem is the special case of (8) with $\alpha = 1$, where the reference model log-likelihood term disappears from the expectation, and the optimal solution is shown by Korbak et al. (2022b) to be

$$\pi^*_{1,\beta}(y \mid x) = p(y \mid x)\exp(\beta r(x,y))/C_{\beta,x}, \tag{9}$$

where $C_{\beta,x} := \int p(y \mid x)\exp(\beta r(x,y))\, dy$ is a normalizing constant.[1] The *tilted distribution* solution of (9) is also known in classical information theory literature (Kullback & Khairat, 1966; Csiszár, 1975). Note that as $\beta \to \infty$, the optimal policy is plain reward maximization over y in the support of the reference model, while for $\beta = 0$, the optimum is just the reference model, i.e., $\pi^*_{1,0} = p$, due to regularization. Sweeping the value of β, these tilted distributions achieve the optimal reward maximization versus KL divergence (from the reference model) trade-off, which is the target for many RL training methods.

Characterizing the general problem of (8) follows as a corollary of the special case of $\alpha = 1$.

Corollary A.1. *The problem given above in* (8) *is equivalently written and solved as*

$$\pi^*_{\alpha,\beta}(y \mid x) = \arg\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\big[\beta r(x,y)\big] - \mathrm{KL}\big(\pi(\cdot \mid x)\|p(\cdot \mid x)^{\alpha}/C_{\alpha,x}\big) \tag{10}$$

$$= \frac{p(y \mid x)^{\alpha}\exp(\beta r(x,y))}{C_{\alpha,\beta,x}}, \tag{11}$$

where $C_{\alpha,x} := \int p(y \mid x)^{\alpha}\, dy$ and $C_{\alpha,\beta,x} := \int p(y \mid x)^{\alpha}\exp(\beta r(x,y))\, dy$ are normalizing constants.

Proof. This can be shown as a corollary of the $\alpha = 1$ special case considered by (Korbak et al., 2022b). However, for completeness, we give the straightforward derivation, starting from (8),

$$\pi^*_{\alpha,\beta}(\cdot \mid x) = \arg\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\big[\beta r(x,y) + (\alpha-1)\log p(y \mid x)\big] - \mathrm{KL}\big(\pi(\cdot \mid x)\|p(\cdot \mid x)\big) \tag{12}$$

$$= \arg\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\left[\beta r(x,y) + \log\frac{p(y \mid x)^{\alpha}}{C_{\alpha,x}\pi(\cdot \mid x)}\right] + \log C_{\alpha,x} \tag{13}$$

$$= \arg\max_{\pi} \mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\frac{p(y \mid x)^{\alpha}\exp(\beta r(x,y))}{C_{\alpha,\beta,x}\pi(\cdot \mid x)}\right] + \log C_{\alpha,\beta,x} \tag{14}$$

$$= \arg\min_{\pi} \mathrm{KL}\left(\pi(\cdot \mid x)\middle\|\frac{p(y \mid x)^{\alpha}\exp(\beta r(x,y))}{C_{\alpha,\beta,x}}\right), \tag{15}$$

where (13) is equivalent to (10), and the final KL minimization is achieved by (11). □

Corollary A.1 states that augmenting the reward with an additional reference model log-likelihood term in (8) is equivalent to adjusting the temperature of the reference model in (10). These equivalent problems are both solved by (11), which we call the *tempered and tilted* distribution, as α is essentially an inverse temperature applied to the reference model, while β controls the exponential tilting towards the reward model. Note that as $\alpha \to \infty$, with a fixed, finite β, the solution approaches the maximum likelihood output of the reference model, whereas $\alpha = 0$ makes the reference model uniform and removes its impact, and values of $\alpha < 0$ inverts the reference model, amplifying less likely outputs.

[1] For discrete $\mathcal{Y}$, the normalizing constants throughout the paper are instead defined via summation.

A.2. Approximate Inference-Time Alignment Techniques

While the tilted (and tempered) distributions of (9) and (11), in principle, express closed-form solutions for the regularized reward maximization problems given by (8) and (10), handling these formulations directly is often computationally intractable. Even with finite sets, where the integration for determining the normalizing constants are replaced with summation, these calculations become infeasible due to the enormous size of the output space (e.g., all token sequences up to a max length). Hence, this subsection reviews some practical inference-time methods that approximate sampling the tilted distribution of (9), i.e., for the special case of $\alpha = 1$, while requiring only the ability to draw independent samples of the reference model and to evaluate the reward model on those samples.

Best-of-N (BoN) is a widely adopted heuristic (see (Stiennon et al., 2020) and (Nakano et al., 2021) for early influential examples) to augment or replace RL training. It simply consists of generating n independent samples, $y_1, \ldots, y_n$, from the reference model $p(y \mid x)$, evaluating each with the reward model, and outputting the sample that maximizes the reward, i.e., $\arg\max_{y_i} r(x, y_i)$. Under certain simplifying assumptions, Yang et al. (2024b) established that BoN asymptotically approaches the optimal tilted distribution. However, in more general finite regimes, which are characterized by Mroueh & Nitsure (2025), BoN does not necessarily approximate the tilted distribution. Best-of-Poisson (BoP) (Khalaf et al., 2025) is an extension of BoN, where, instead of a fixed n, the number of samples is drawn from a Poisson distribution, which results in a close approximation of the tilted distribution, under the assumption of uniformly distributed rewards.

Soft Best-of-N (SBoN) (Verdun et al., 2025) similarly generates n independent samples and evaluates the reward, $r_i = r(x, y_i)$, for each sample. However, instead of selecting the maximizer, SBoN randomly selects sample y_i with probability $\exp(\beta r_i)/\sum_j \exp(\beta r_j)$, i.e., the distribution over the n samples formed by the softmax of their rewards, with inverse temperature β. Verdun et al. (2025) characterize the accuracy of SBoN, which includes establishing that $\mathrm{KL}(\pi^*_{1,\beta} \| \tilde{\pi}_{\beta,n}) = O(1/n)$, where $\tilde{\pi}_{\beta,n}$ denotes the distribution sampled by SBoN. Reward Augmented Decoding (RAD) (Deng & Raffel, 2023) employs a similar concept, but, instead of selecting among entire response candidates, RAD utilizes the reward model to augment the sampling of each token, within the iterative process of auto-regressive generation.

Jinnai et al. (2025) propose methods to augment the reward model for regularizing BoN to mitigate reward-hacking. Their main proposal, inspired by minimum Bayes risk decoding, involves a proximity regularizer that aims to minimize the Wasserstein distance with respect to the reference policy. As an ablation, Jinnai et al. (2025) also consider what they call KL-regularized BoN, which essentially augments the reward with the reference model log-likelihood, and is analogous to the $(\alpha - 1) \log p(y \mid x)$ term in the tempered reward maximization of (8).

A.3. Sharpened Logarithmic Opinion Pooling

The proxy reward model r implicitly defines the *alignment distribution*, given by

$$q(y \mid x) := \exp(r(x, y))/R_x, \tag{16}$$

where $R_x := \int \exp(r(x, y))\ dy$ is a normalizing constant. The tempered and tilted distribution, given in (11), can be rewritten as

$$\pi^*_{\alpha,\beta}(y \mid x) = \frac{p(y \mid x)^\alpha q(y \mid x)^\beta}{\int p(z \mid x)^\alpha q(z \mid x)^\beta\ dz} \tag{17}$$

$$=: \mathrm{softmax}\big(\alpha \log p(y \mid x) + \beta \log q(y \mid x)\big), \tag{18}$$

where $\mathrm{softmax}(\cdot)$ denotes exponentiation and normalization, such that we have valid distributions over $y \in \mathcal{Y}$, for each $x \in \mathcal{X}$. This form makes it clear that the tempered and tilted distribution is essentially a *logarithmic opinion pool* (LOP) (Genest & Zidek, 1986; Heskes, 1997), which is also related to the concept of *product of experts* (Hinton, 2002). However, unlike literature that typically considers LOP with non-negative weights that sum to one, we instead employ arbitrary $\alpha, \beta \in \mathbb{R}$, which allows for concentrating probability mass on the most confident outputs (e.g., converging towards proxy reward maximization as $\beta \to \infty$) or even inverting the contribution of anti-aligned models (e.g., if the negated proxy reward is closer to the gold reward, then $\beta < 0$ may be more effective). We emphasize this aspect by calling this form of model as a *Sharpened LOP* (SLOP).[2]

Inspired by this logarithmic pooling perspective, we can also generalize to a pool of m experts (i.e., generative and/or proxy reward models), denoted by $s_1, \ldots, s_m$, representing the score contributed by each expert, where $s_i(x, y) = \log p_i(y \mid x)$

[2]Admittedly, also for the sake of utilizing a memorable and ironic acronym.

for a generative model and $s_i(x, y) = r_i(x, y) \equiv \log q_i(y \mid x)$ for a reward model.[3] These experts are combined, with weights $\omega := (\omega_1, \ldots, \omega_m) \in \mathbb{R}^m$, to form the SLOP given by

$$\pi^*_\omega(y \mid x) = \text{softmax}\left(\sum_{i=1}^{m} \omega_i s_i(x, y)\right) \propto \prod_{i=1}^{m} p_i(y \mid x)^{\omega_i}, \tag{19}$$

where, for notational convenience, each p_i denotes either a generative model or the corresponding alignment distribution q_i of a reward model.

Directly sampling the SLOP distribution from (19) is often intractable, when the output space $\mathcal{Y}$ is very large. However, (19) can also be interpreted as the solution of a regularized reward maximization problem, where the proxy reward is a weighted ensemble of expert scores, which allows us to approximately sample from the SLOP via the following extension of SBoN (Verdun et al., 2025). By convention, we set the generative reference model as the first expert p_1. For a given prompt x, we sample n candidates $y_1, \ldots, y_n \overset{\text{iid}}{\sim} p_1(\cdot \mid x)$, evaluate candidate rewards as function of expert scores, as given by,

$$r'(x, y_j) := (\omega_1 - 1) \log p_1(y_j \mid x) + \sum_{i=2}^{m} \omega_i s_i(x, y_j), \tag{20}$$

and select one candidate according to

$$\widehat{\pi}_{\omega,n}(y_j \mid x) = \text{softmax}(r'(x, y_j)) \propto \exp\left(\sum_{i=1}^{m} \omega_i s_i(x, y_j) - s_1(x, y_j)\right), \tag{21}$$

where $\text{softmax}(\cdot)$ denotes exponentiation and normalization over the set of n candidates. Note that the first expert weight is set to $(\omega_1 - 1)$, which accounts for the candidates being already sampled from the first expert.

Corollary A.2. *Let $\widehat{\pi}_{\omega,n}$ denote the distribution realized via the above extension of SBoN, by selecting from n sampled candidates $y_1, \ldots, y_n \overset{iid}{\sim} p_1(\cdot \mid x)$, according to (21). This approximates sampling the SLOP distribution π^*_ω given in (19), with KL$(\pi^*_\omega \| \widehat{\pi}_{\omega,n}) = O(1/n)$.*

Proof. This corollary follows immediately from Theorem 1 of (Verdun et al., 2025), which shows that $\text{KL}(\pi \| \widehat{\pi}_{\omega,n}) = O(1/n)$, for the distribution $\pi \propto p_1 \exp(r')$, since $\widehat{\pi}_{\omega,n}$ is produced by essentially applying SBoN with the reference model p_1 and the reward model r' given by (20). Expanding r', we have $p_1 \exp(r') = \prod_{i=1}^{m} \exp(\omega_i s_i)$, and thus $\pi = \pi^*_\omega$. □

In the special case of one reference model and one reward model, this approximates tempered-and-tilted sampling, with candidate likelihoods proportional to $\exp\big((\alpha - 1) \log p(y_j \mid x) + \beta r(x, y_j)\big)$, where (α, β) correspond to (ω_1, ω_2).

A.4. Reward Hacking Mitigation via Calibrated Model Pooling

The motivation for considering regularized reward maximization is that neither the reference model p nor the (proxy) reward model r is perfect. The ultimate objective is to attain a policy that maximizes a gold reward or closely approximates an ideal distribution, neither of which may be readily available or precisely known. Intuitively, optimizing toward an inaccurate proxy reward model r may be misleading and counter-productive to the ultimate objective. This is a well-known phenomenon commonly referred to as *reward hacking* (Pan et al., 2022), for which the recent works of (Mroueh & Nitsure, 2025; Huang et al., 2025; Aminian et al., 2025; Khalaf et al., 2025) have provided theoretical characterizations and related mitigation strategies. In particular, our approach extends upon the HedgeTune concept of (Khalaf et al., 2025) by utilizing a small amount of calibration samples to optimize the SLOP weights ω.

The ultimate objective (in principle) of our reward hacking mitigation approach is to maximize the expected gold reward, over the choice of SLOP weight parameters, as given by

$$\sup_{\omega} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi^*_\omega(y|x)}[g(x, y)], \tag{22}$$

where the expectation is over both sampling x from some input distribution $\mathcal{D}$, and sampling y from the SLOP π^*_ω. We remark that this optimization problem does not always have a maximizer with finite weights. To illustrate, consider the

[3] Due to later normalization via softmax, the normalizing constant of the alignment distribution may be omitted.

following simple binary example, where $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, with $g(x, y) = 2|y - x|$, input distribution $\mathcal{D} = \text{Bernoulli}(0.5)$, and a single expert ($m = 1$) SLOP,

$$\pi^*_\omega(y \mid x) = \text{softmax}(\omega_1 \log p_1(y \mid x)) = \rho\left(\omega_1 \log \frac{p_1(y \mid x)}{p_1(1 - y \mid x)}\right), \tag{23}$$

where $\rho(z) := 1/(1 + \exp(-z))$ is the sigmoid function, and we assume that $p_1(y \mid x) \in (0, 1)$ for all $x, y \in \{0, 1\}$. For this example, the expected reward is given by

$$\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi^*_\omega(y|x)}[g(x, y)] = \rho\left(\omega_1 \log \frac{p_1(1 \mid 0)}{p_1(0 \mid 0)}\right) + \rho\left(\omega_1 \log \frac{p_1(0 \mid 1)}{p_1(1 \mid 1)}\right), \tag{24}$$

as a function of two log-likelihood ratios. If both ratios are zero, then the expected reward equals one for any ω_1. If one ratio is strictly positive and the other is strictly negative, then a unique maximum exists for some finite ω_1. However, if both ratios have the same sign, or if only one of the ratios is zero, then the supremum of the expected reward (which equals two, if they are both non-zero, or 1.5, if one ratio is zero) is only approached as $\omega_1 \to \infty$ (if the ratios are non-negative) or $\omega_1 \to -\infty$ (if the ratios are non-positive). We can interpret these latter cases as the model p_1 being universally aligned or anti-aligned (across all inputs $x \in \{0, 1\}$), and divergence of the weight ω_1 to $\pm\infty$ has the effect of sharpening the model towards a hard (deterministic) output.

In practice, we generally cannot arbitrarily evaluate the gold reward, which limits us to empirically estimate the objective in (22) via a small number of calibration samples. We assume that we are given k input samples, $x_1, \ldots, x_k$, drawn iid from the input distribution $\mathcal{D}$, and that for each x_i, we can query the gold reward for up to n samples. In practice, for situations with a verifiable reward, instead of querying the gold model up to kn times, this assumption may instead be realized by just having the gold reference responses corresponding to the k input samples (e.g., for math problems, responses can be checked against reference solutions to assign a correctness reward). Algorithm 1 uses these calibration samples to calculate the following empirical estimate of the objective in (22), upon which it performs gradient ascent, as given by

$$\widehat{J}(\omega) = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} \widehat{\pi}_{\omega,n}(y_{ij} \mid x_i) g(x_i, y_{ij}) - \lambda \|\omega\|^2, \tag{25}$$

where $\lambda \geq 0$ is optionally used to regularize weight optimization and avoid divergence towards fully sharpened hard outputs. Given calibrated weights ω, we can either approximately sample the SLOP distribution via the method given by (21), or instead employ a hard selection that simply picks the candidate y_j that maximizes the weighted SLOP score $\sum_i \omega_i s_i(x, y_j) - s_1(x, y_j)$.

B. Visual Question Answering Experiment Details and Additional Results

Figures 4 and 5 plot additional performance results for Gemma-3-4B (Gemma Team et al., 2025) and Qwen3-VL-4B (Bai et al., 2025), while Figure 6 plots their corresponding optimized SLOP weights.

The VLM is prompted with image and question text pairs from the SQA dataset, preceded by the following system prompt:

```
A chat between a curious user and an artificial intelligence assistant.
The assistant gives helpful answers to the user's multiple-choice questions,
responding with the letter of the correct choice only.
```

Weight optimization is performed for $T = 500$ steps, with learning rate $\eta = 0.05$ and weight decay parameter $\lambda = 0$, and by using Adam (Kingma & Ba, 2014), with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

For this experiment, we restrict sampling to only decoding a single-token over the small subset of possible answers, which allows for exactly sampling the corresponding SLOP (instead of approximate sampling via SBoN). This simplifies the calibration objective in (7) to

$$\widehat{J}(\alpha, \beta) = \frac{1}{k} \sum_{i=1}^{k} \sum_{y \in \mathcal{Y}(x_k)} \text{softmax}\big(\alpha \log p(y \mid x_k) + \beta r(x_k, y)\big) g(x_i, y), \tag{26}$$

where each inner summation and softmax is only over $\mathcal{Y}(x_i) \subset \mathcal{Y}$, which denotes the subset of possible answers provided by each multiple-choice question x_i.

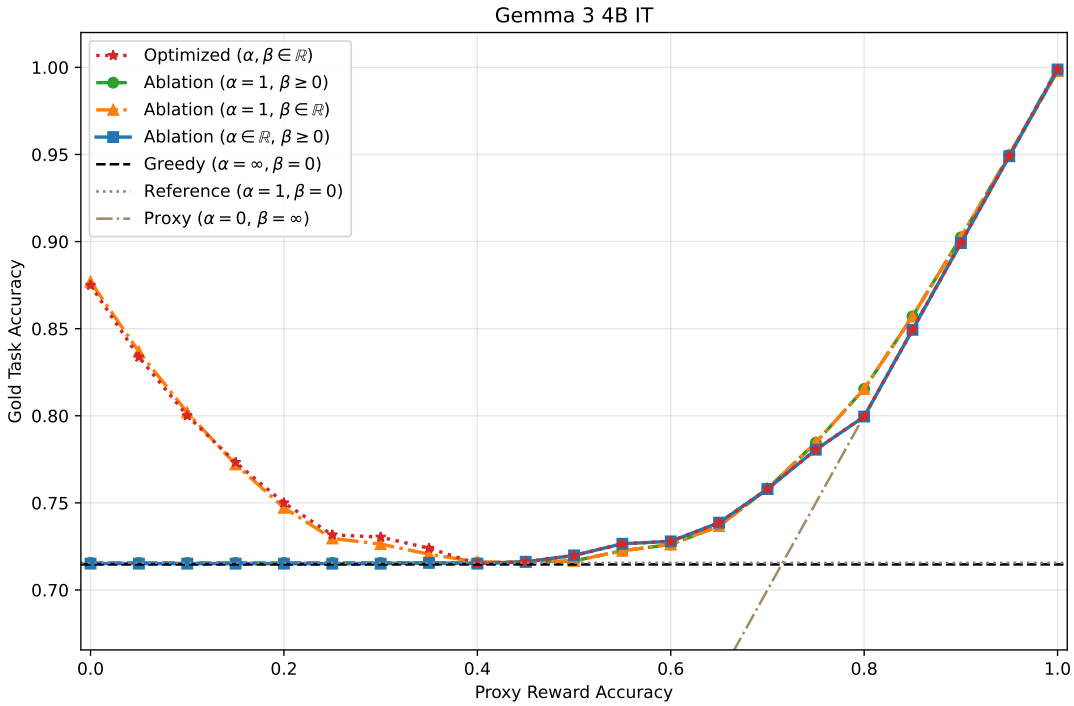

Figure 4. SLOP with Gemma-3-4B paired with synthetic proxy reward with varying accuracy, evaluated on SQA.

C. Math Reasoning Experiment Details

Responses are generated for up to 1024 tokens, with the following system instruction, preceding each GSM8K question provided as the prompt:

```
You are a careful math tutor. Solve the user's grade-school math problem,
show your reasoning, and end with 'Final answer: <number>'.
```

To check the correctness of a response, the answer is extracted using a regular-expression to find a number prefixed by "Final answer:", while falling back to extracting the last number in the response, if the prefix is not found, and then checking against the reference answer via string matching.

Weight optimization is performed for $T = 500$ steps, with learning rate $\eta = 0.05$ and weight decay parameter $\lambda = 10^{-5}$, and by using Adam, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

For convenience, Figures 7, 8, 9, and 10 reproduce the earlier Figure 3, but in larger format.

D. Datasets and Models

For the visual question answering experiments in Section 3.1 and Appendix B, we used the following dataset and VLMs:

- ScienceQA (Lu et al., 2022):
  - License: Creative Commons Attribution Share Alike 4.0 (CC-BY-SA-4.0)
  - https://huggingface.co/datasets/derek-thomas/ScienceQA
- LLaVA-1.5-7B (Liu et al., 2023; 2024):
  - License: Llama 2 Community License Agreement
  - https://huggingface.co/llava-hf/llava-1.5-7b-hf
- Gemma-3-4B (Gemma Team et al., 2025):

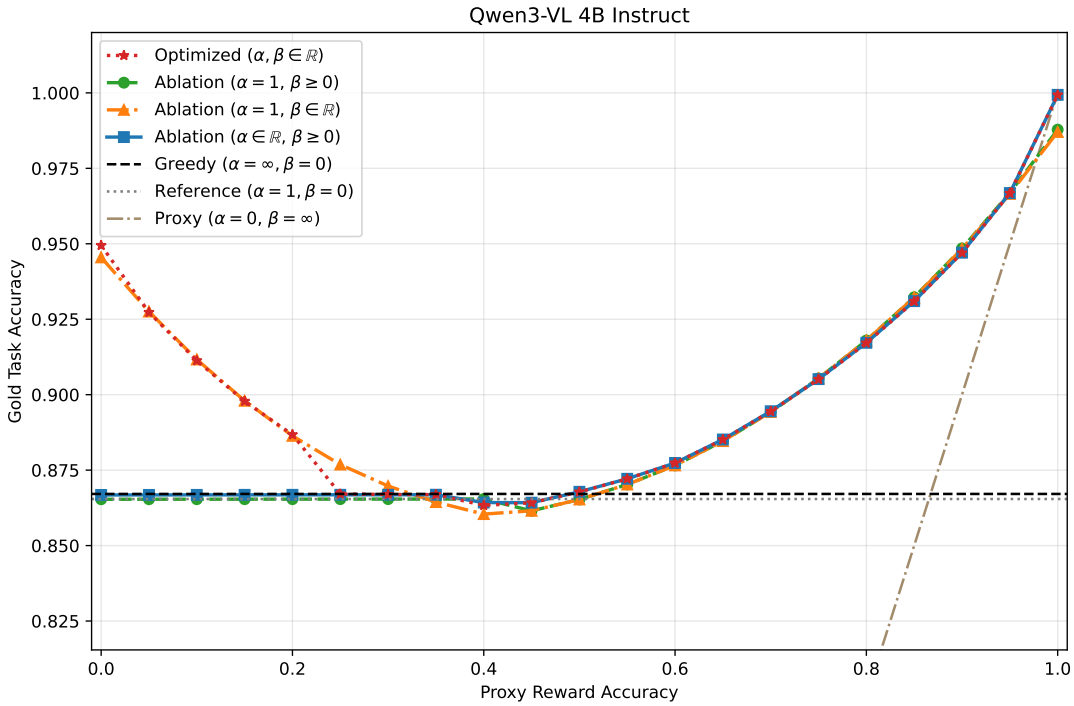

Figure 5. SLOP with Qwen3-VL-4B paired with synthetic proxy reward with varying accuracy, evaluated on SQA.

  - License: Gemma Terms of Use
  - https://huggingface.co/google/gemma-3-4b-it

- Qwen3-VL-4B (Bai et al., 2025):
  - License: Apache License 2.0
  - https://huggingface.co/Qwen/Qwen3-VL-4B-Instruct

For the math reasoning experiments in Section 3.2 and Appendix C, we used the following dataset and LLMs:

- GSM8K (Cobbe et al., 2021):
  - License: MIT License
  - https://huggingface.co/datasets/openai/gsm8k
- Gemma-3-1B (Gemma Team et al., 2025):
  - License: Gemma Terms of Use
  - https://huggingface.co/google/gemma-3-1b-it
- Qwen2-1.5B (Yang et al., 2024a):
  - License: Apache License 2.0
  - https://huggingface.co/Qwen/Qwen2-1.5B-Instruct
- Qwen3-0.6B (Yang et al., 2025):
  - License: Apache License 2.0
  - https://huggingface.co/Qwen/Qwen3-0.6B
- Phi-3.5-mini (Abdin et al., 2024):

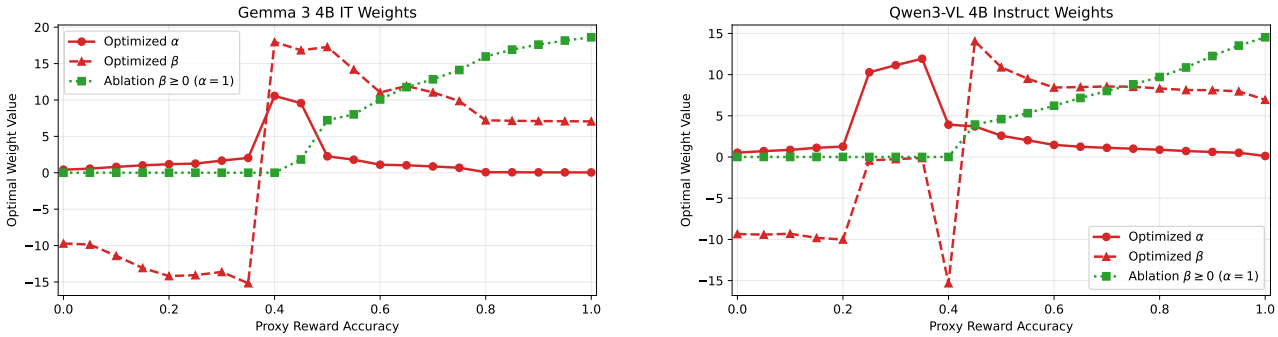

Figure 6. Optimized (two-expert) SLOP weights for SQA.

- License: MIT License
- https://huggingface.co/microsoft/Phi-3.5-mini-instruct

E. Computational Resource Usage

All experiments were run on a single desktop workstation, equipped with an Intel Core i7-13700K CPU, 128 GB of system RAM, and an Nvidia RTX 4090 GPU with 24 GB of VRAM. The storage footprint for all models, datasets, and results generated by our experiments is approximately 48 GB.

In total, our experiments used approximately 150 to 200 hours of compute on this machine. The vast majority of this time was used by the math reasoning experiments, discussed in Section 3.2 and Appendix C, specifically response generation and scoring. The visual question answering experiments of Section 3.1 and Appendix B consisted of no more than 3 hours of the total compute, since our approach for these multiple-choice questions required only computing single next-token likelihoods.

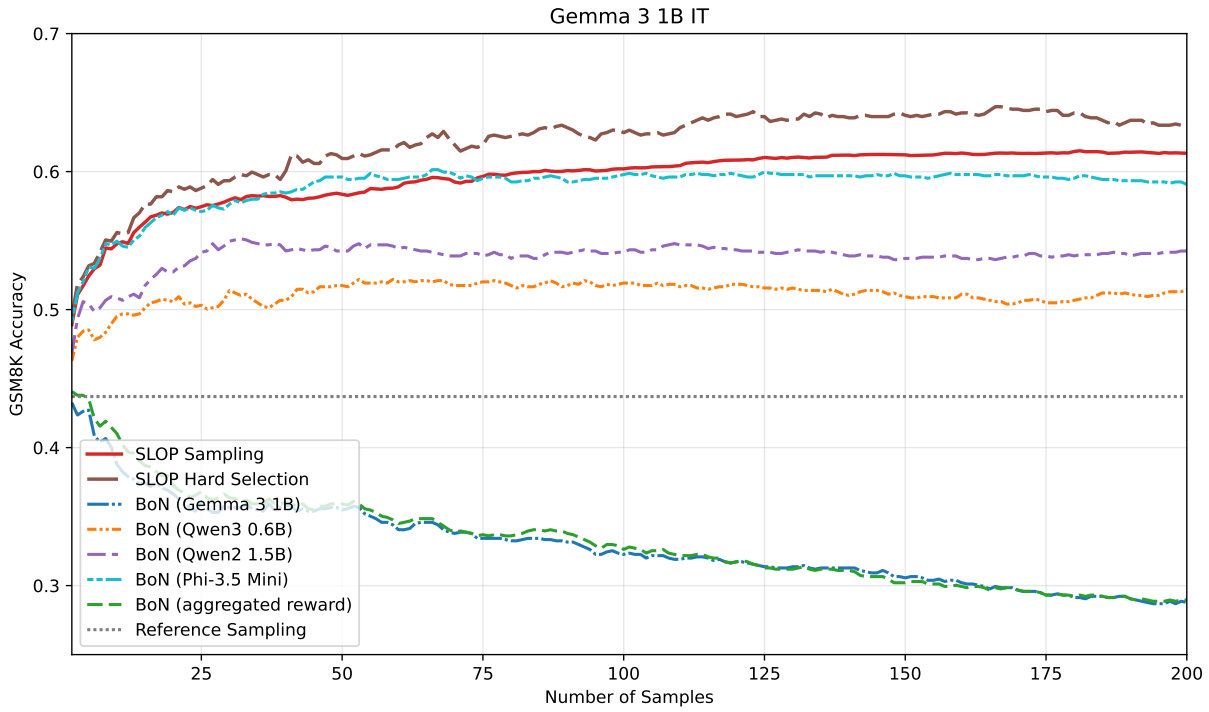

Figure 7. SLOP (with 4 LLMs) evaluated on GSM8K, with Gemma-3-1B as the reference model.

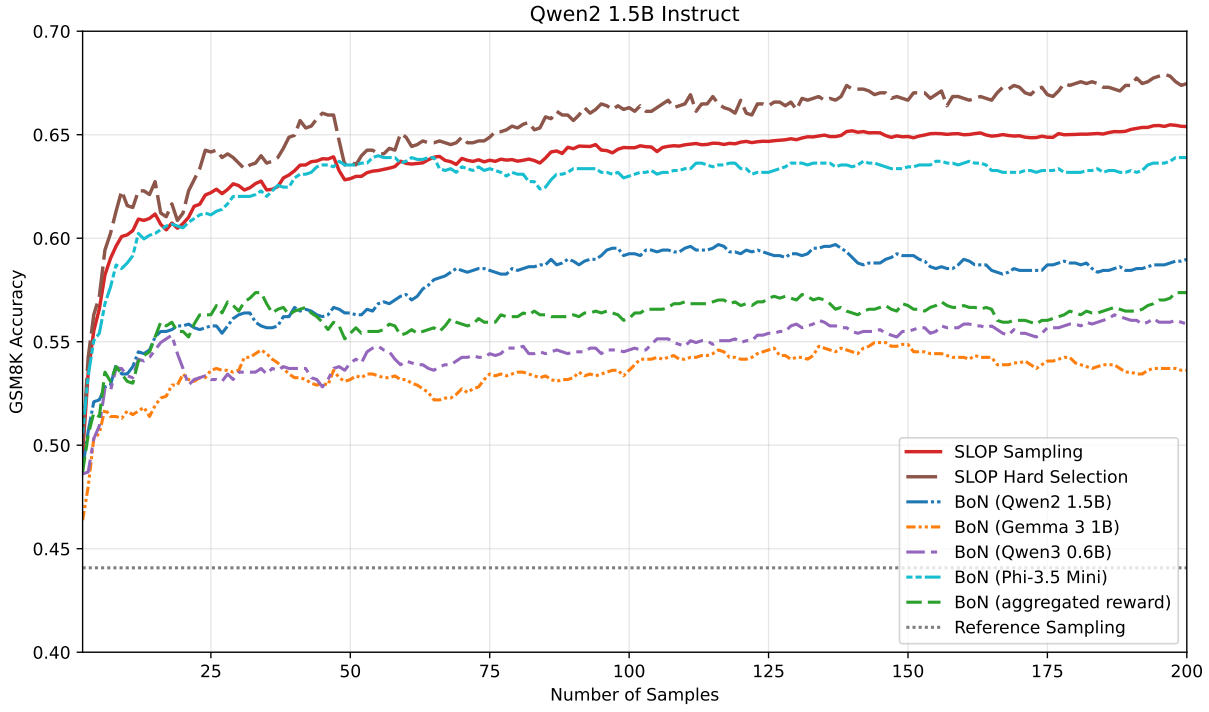

Figure 8. SLOP (with 4 LLMs) evaluated on GSM8K, with Qwen2-1.5B as the reference model.

Figure 9. SLOP (with 4 LLMs) evaluated on GSM8K, with Qwen3-0.6B as the reference model.

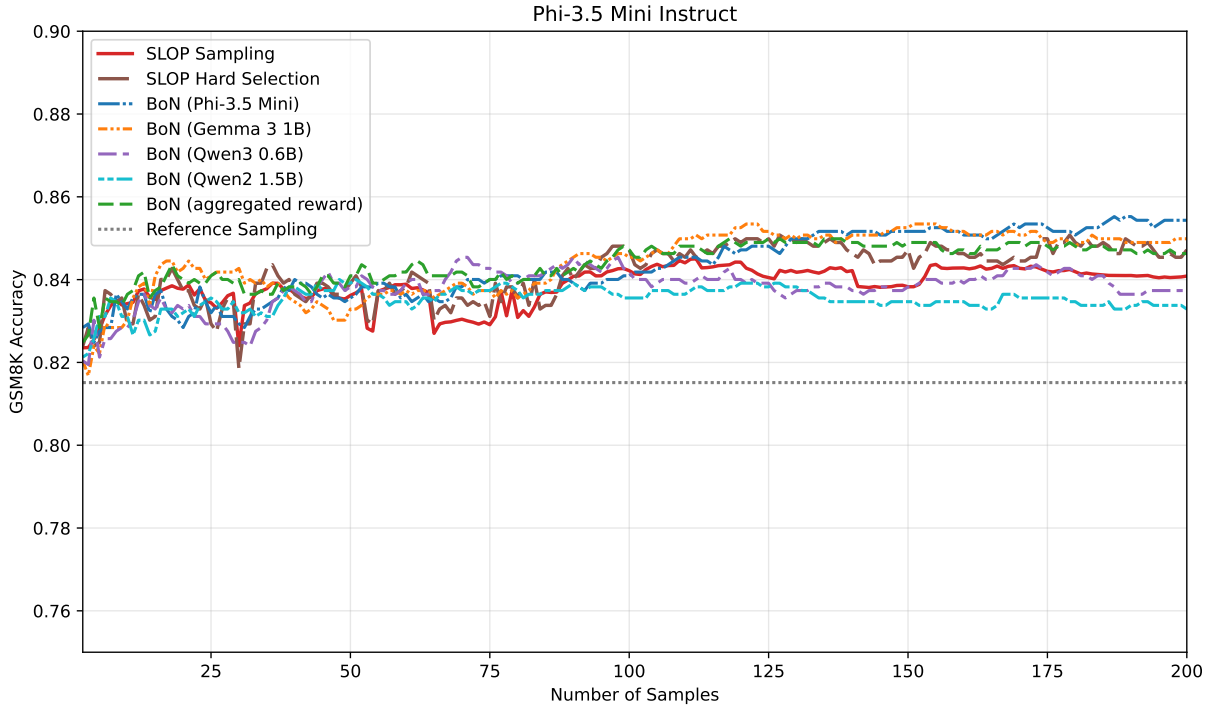

Figure 10. SLOP (with 4 LLMs) evaluated on GSM8K, with Phi-3.5-mini as the reference model.