

On the Rejection Criterion for Proxy-based Test-time Alignment

Ayoub Hammal¹ Pierre Zweigenbaum¹ Caio Corro²

¹Université Paris-Saclay, CNRS, LISN

²INSA Rennes, IRISA, CNRS, Université de Rennes

{ayoub.hammal,pz}@lisn.fr caio.corro@irisa.fr

Abstract

Recent work proposed test-time alignment methods that rely on a small aligned model as a proxy that guides the generation of a larger base (unaligned) model. The implicit reward approach skews the large model’s distribution (Mitchell et al., 2024), whereas the nudging approach defers the generation of the next token to the small aligned model when the large base model is unconfident about its outcome (Fei et al., 2025; Hammal et al., 2026). In this work, we first show that both approaches can be reduced to sampling from similar graphical models, differing only in the definition of a rejection criterion (or distribution). Moreover, we argue that the confidence criterion is ill-motivated due to linguistic phenomena like ambiguous phrasing. We propose a novel rejection criterion based on a *conservative confidence bet*. Experimentally, our novel approach outperforms previous work on several datasets.

1 Introduction

Recently, large language models (LLMs) have been the main focus of the natural language processing research community. Training an LLM is a costly and difficult process, which can be summarized into the following three phases:

1. *pretraining*, where the language model is trained on a large collection of texts;
2. *supervised fine-tuning* (SFT), where the model is trained on selected input/output pairs, for example to learn to follow instructions;
3. *preference learning*, which encompasses a diverse set of extra learning steps including reinforcement learning from human feedback (RLHF), direct preference optimization (DPO), and reinforcement learning from verifiable rewards (RLVR).

Phases 2 and 3, which are often referred to as *alignment* phases, are essential to bias LLMs so that they follow expected (human) preferences, formatting or instructions, and more generally any downstream task requirements. Unfortunately, this procedure is costly and scales poorly with the model size (Lambert et al., 2025).

As such, test-time alignment has been proposed as an alternative to the alignment phases. In this setting, the distribution of a base model is biased directly during the generation process. A first approach is to rely on an explicit reward. If the reward is partial (or unidirectional, *i.e.*, it can be applied on unfinished generations), it is then possible to directly select next token candidates by using the LLM and the reward in a pipeline (Deng and Raffel, 2023; Cao et al., 2023; Khanov et al., 2024, *inter alia*). However, depending on the reward architecture, partial rewards can be costly to use as they require one forward pass per next token candidate (Deng and Raffel, 2023) or are of limited expressivity (Troshin et al., 2025), and they are difficult to train (Rashid et al., 2025). Otherwise, for rewards defined on finished generations only, previous work proposed to rely on generation reranking (Gao et al., 2023; Li et al., 2023) and Markov-chain Monte Carlo sampling (Faria and Smith, 2025). An important downside of these methods is that they require drawing many samples from the base LLM, which means the generation process is very slow.

A second approach is to bias a large base LLM with a small aligned LLM, which we call proxy-based test-time alignment in the following. A popular approach is to extract an *implicit partial reward* from the small aligned model, and use it to “emulate” the distribution of the (unknown) large aligned LLM from its base version (Mitchell et al., 2024; Liu et al., 2024; Li et al., 2025). However, building the implicit reward requires access to both the base and aligned versions of the small model. Recently, Fei et al. (2025) proposed the *nudging* method: at

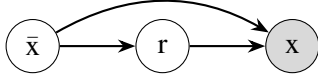


Figure 1: Probabilistic graphical model of the distribution π for proxy-based test-time alignment.

each generation step, they either generate the next token from the base large model if it has a low min-entropy, or otherwise from the aligned small model. Hammal et al. (2026) extended nudging to token-specific decision, *i.e.*, they propose a framework in which the decision to defer generation to the aligned small model depends on the token sampled from the large one. However, their approach is based on the same intuition: the generation should be deferred to the aligned model when the base model is *unconfident about its outcome*.

Contributions. In this work, we introduce a new graphical model for proxy-based test-time alignment that can be parametrized through a *rejection criterion* (or distribution). We show that both implicit reward and nudging approaches can be modeled in this framework, showing that they are related, although very different at first glance. Moreover, we argue that the confidence criterion of nudging is ill-motivated due to linguistic phenomena like ambiguous phrasing. We propose instead a novel rejection criterion based on a *conservative confidence bet*. Experimentally, our novel approach outperforms previous work on several datasets. Code to reproduce experiments is publicly available.¹

Notations. We write p and q the large and small language models, respectively. We will distinguish the aligned version of each model as p^* and q^* . The token vocabulary is the set $V = \{1, \dots, |V|\}$. To simplify notation, we drop dependence on the generation context, and we note the next token probability vector with bold letters such as \mathbf{p} , indexed by the vocabulary, that is the probability of token $v \in V$ is written p_v . Note that $\mathbf{p} \in \Delta(|V|)$ where $\Delta(k)$ is the $(k-1)$ -dimensional simplex. Finally, we write Iverson brackets as $\llbracket \cdot \rrbracket$, which evaluates to 1 if the inner condition is true and 0 otherwise.

2 Rejection-based Test-time Alignment

In this section, we show that previous work on proxy-based test-time alignment can be modeled

¹<https://github.com/ayoubhammal/knapsack-approximation-deferral>

as sampling from a *probabilistic graphical model* (PGM), where a rejection criterion decides whether a sample from the base large model \mathbf{p} should be used as the next token or not.

Graphical model. Let π be a PGM on the following random variables (RV):

- \bar{x} , a latent RV on the vocabulary;
- r , a latent binary RV whose samples indicate whether the \bar{x} sample should be rejected ($r = 1$) or not ($r = 0$);
- x , the observed RV over the vocabulary (*i.e.*, the generated token).

The generative story is defined as follows:

1. Sample a draft token w from $\pi(\bar{x})$;
2. Sample a decision from the rejection distribution given the token w sampled before, *i.e.*, $r \sim \pi(r | \bar{x} = w)$;
3. Sample a final token $v \sim \pi(x | r = r, \bar{x} = w)$, where we either copy the draft token if $r = 0$ or sample a new token otherwise.

This PGM is depicted in Figure 1, and the resulting distribution has the following form:

$$\begin{aligned} \pi(x = v) &= \sum_{w \in V} \pi(\bar{x} = w) \pi(r = 0 | \bar{x} = w) \\ &\quad \times \pi(x = v | r = 0, \bar{x} = w) \\ &\quad + \sum_{w \in V} \pi(\bar{x} = w) \pi(r = 1 | \bar{x} = w) \\ &\quad \times \pi(x = v | r = 1, \bar{x} = w). \end{aligned}$$

The latent token distribution is set to the large base model distribution:

$$\pi(\bar{x} = w) = p_w,$$

and we sample from the aligned small model if the draft sample is rejected, that is:

$$\pi(x = v | r = r, \bar{x} = w) = \begin{cases} \llbracket v = w \rrbracket & \text{if } r = 0, \\ q_v^* & \text{otherwise.} \end{cases}$$

As such, we are left with only defining the rejection distribution $\pi(r | \bar{x})$, that is, defining the Bernoulli parameters $\mu_v \in [0, 1]$:

$$\pi(r = 1 | \bar{x} = v) = \mu_v.$$

Nudging and dual KAD. On the one hand, the nudging method of Fei et al. (2025) corresponds to using the following rejection distribution:

$$\pi(\mathbf{r} = 1 \mid \bar{\mathbf{x}} = v) = \llbracket \max_w p_w < \lambda \rrbracket,$$

where $\lambda \in [0, 1]$ is a thresholding hyperparameter on the distribution confidence. Interestingly, this rejection criterion does not depend on the initial draft token v , that is:

$$\pi(\mathbf{r} = 1 \mid \bar{\mathbf{x}} = v) = \pi(\mathbf{r} = 1).$$

This means the rejection decision is a distribution-level decision in this case, *i.e.*, we reject the sampling distribution \mathbf{p} , not the exact sampled token.

On the other hand, the dual version of knapsack approximate deferral (KAD, Hammal et al., 2026) corresponds to using the following distribution:

$$\pi(\mathbf{r} = 1 \mid \bar{\mathbf{x}} = v) = \llbracket p_v < \lambda \rrbracket.$$

In this case, the token v is rejected if \mathbf{p} is not confident enough in the probability of v .

Note, however, that both of these rejection criteria are deterministic, that is:

$$\pi(\mathbf{r} = 1 \mid \bar{\mathbf{x}} = v) \in \{0, 1\}, \quad \forall v \in V.$$

Implicit reward. Implicit reward alignment (Mitchell et al., 2024; Liu et al., 2024; Li et al., 2025) relies on extracting a token-level reward component as the ratio q^*/q , which represents the knowledge learned by q^* during alignment. This partial reward is then used to bias p with the extracted alignment information, constructing a new aligned distribution \mathbf{s} defined as follows:

$$s_v = p_v \frac{q_v^*}{q_v Z}, \quad \text{with} \quad Z = \sum_w p_w \frac{q_w^*}{q_w},$$

where Z is for normalization, *i.e.*, to ensure that $\mathbf{s} \in \Delta(|V|)$. Although this approach seems different from the nudging and KAD, the following proposition shows it can also be reduced to defining a rejection criterion in our PGM.

Proposition 1. *Let \mathbf{p} , \mathbf{q} and \mathbf{q}^* be probability distributions over the vocabulary such that there exists $\alpha \in [0, 1]$ with*

$$\forall v \in V : \quad \frac{s_v - p_v}{q_v^*} \leq \alpha \leq \frac{s_v}{q_v^*},$$

or, alternatively,

$$\forall v \in V : \quad q_v^* \alpha \leq s_v \leq p_v + q_v^* \alpha. \quad (1)$$

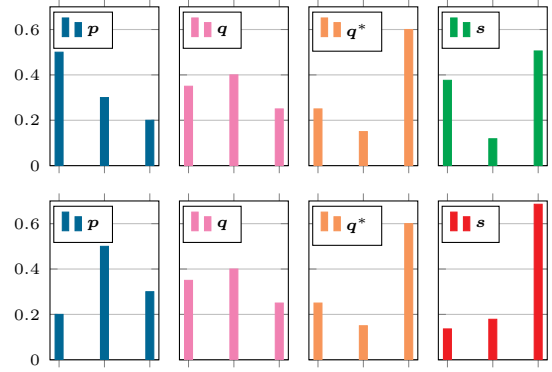


Figure 2: Two examples of implicit reward mixture distributions. The upper row shows an example of \mathbf{p} , \mathbf{q} and \mathbf{q}^* that produce a mixture \mathbf{s} for which an α as defined in Proposition 1 exists. The lower row shows an example of the opposite case, where no such α exists. Note that both cases differ only in distribution \mathbf{p} , which is “unaligned”, *i.e.*, the most probable token in \mathbf{p} is different from the most probable token in \mathbf{q}^* .

Then, the rejection distribution defined as

$$\forall v \in V : \quad \pi(\mathbf{r} = 1 \mid \bar{\mathbf{x}} = v) = \frac{p_v + q_v^* \alpha - s_v}{p_v},$$

gives the following PGM distribution:

$$\forall v \in V : \quad \pi(\bar{\mathbf{x}} = v) = s_v.$$

Proof is given in Appendix A. Proposition 1 gives a sufficient condition on \mathbf{p} , \mathbf{q} and \mathbf{q}^* so that the implicit reward distribution \mathbf{s} can be cast in our PGM, which reduces to the “enclosing” constraint in Equation 1. Note that this proposition suggests that, in some cases, our PGM cannot represent the distribution \mathbf{s} .

Figure 2 shows distributions that either satisfy or do not satisfy the condition.

3 What is a Good Rejection Criterion?

Nudging and KAD rely only on p to take the rejection decision. This is based on the assumption that the fallback distribution is an expert distribution whose error can be ignored (Chow, 1970). However, in practice, q^* can underperform p . Considering information from q^* can yield a richer rejection decision.

Moreover, relying on the absolute confidence of p alone is a bad criterion to assess the quality of a generation (Dinh and Niehues, 2025). In many cases, probability mass can be split across several correct tokens, which reduces the overall probability assigned to each of them. Although this appears

	GMS8K MATH SVAMP ARC CSQA					Avg.	GMS8K MATH SVAMP ARC CSQA					Avg.
	OLMo 2 (1B and 13B)						Qwen 3 (1.7B and 14B)					
Building blocks												
p	54.5	09.4	57.6	29.6	19.4	34.1	75.5	51.8	80.0	86.6	76.9	74.1
q^*	62.5	16.4	70.3	43.8	48.4	48.2	75.3	53.0	86.6	82.9	68.7	73.3
Previous works												
Imp. reward	58.4	18.2	73.0	63.3	<u>55.8</u>	53.7	80.7	60.6	89.0	88.9	78.1	79.4
↔ Nudging (Fei et al., 2025)												
$\lambda=0.3$	60.3	13.8	67.3	53.7	33.4	45.7	78.1	52.8	85.0	88.8	78.5	76.6
$\lambda=0.4$	61.9	18.8	66.6	57.8	46.3	50.2	79.5	57.8	87.3	89.6	79.3	78.7
↔ Dual KAD (Hammal et al., 2026)												
$\lambda=0.3$	69.5	22.8	76.6	61.6	52.3	56.5	81.3	59.6	89.0	<u>91.4</u>	81.3	80.5
$\lambda=0.4$	72.3	23.4	75.3	61.9	55.6	57.7	81.7	60.6	87.3	91.5	<u>80.7</u>	<u>80.3</u>
Our work: confidence bet												
$\lambda=0$	71.1	21.4	77.6	56.6	53.4	56.0	78.3	61.8	90.3	87.3	74.7	78.4
$\lambda=0.1$	70.5	24.0	79.3	59.5	58.7	<u>58.4</u>	82.4	59.8	89.0	89.5	79.2	79.9
$\lambda=0.2$	<u>71.7</u>	26.4	<u>79.0</u>	<u>62.6</u>	54.9	58.9	<u>82.1</u>	<u>61.6</u>	<u>89.3</u>	90.5	79.3	80.5
For reference												
p^*	84.3	39.6	87.6	82.5	76.9	74.1	82.4	64.0	88.3	93.8	83.1	82.3
q	14.4	01.0	17.6	19.6	15.8	13.6	21.9	12.2	25.3	47.1	29.5	27.2

Table 1: Accuracies given by different models and methods. Best results on each column are in bold, second best are underlined.

to indicate the model is unconfident, it actually shows ambiguous phrasing and related linguistic phenomena. A naive example can be seen in splitting the model probability mass between “like” and “such”, which are both valid introductions of examples (“frameworks like Pytorch” vs “frameworks such as Pytorch”). As both those continuations are valid, a uniform probability split would result in ≤ 0.5 probability mass for each of these options, e.g., $p_{\text{that}} = 0.35$, $p_{\text{like}} = 0.35$ and the rest of the mass is spread on other tokens, all of them having very low probabilities. This case corresponds to natural ambiguity in the language, and not to a lack of confidence of the model *per se*. As such, we argue that confidence-based rejection criteria are ill-motivated.

Instead of using fixed thresholds, as in nudging and KAD’s dual rule, we can intuitively compare the initial draft from p to the best possible guess from q^* . In other words, we introduce a baseline to which we compare the confidence of the prediction from p , and we additionally avoid rejecting the initial draft from p if we have guarantees that q^* cannot improve it.

Conservative confidence bet. A rational actor drawing an initial draft v from p would compare the confidence of this draft p_v to the best confidence that q^* can produce, that is, to $\max_w q_w^*$. A

decision to keep this draft can be taken safely if its confidence p_v is larger than that of the best possible sample from q^* . In the opposite case, a decision to defer is taken instead. This results in a deferral rule that we call *conservative confidence bet*, which is defined as follows:

$$\pi(r = 1 \mid \bar{x} = v) = \llbracket p_v < \max_w q_w^* - \lambda \rrbracket,$$

where $\lambda \in [0, 1]$ is a margin hyperparameter. Note that this rule mimics deferral rules previously proposed for cascading (Narasimhan et al., 2025).

4 Experiments

Datasets and models. We use the same experimental setup as Hammal et al. (2026). We select three *math reasoning* datasets, GSM8K (Cobbe et al., 2021), MATH500 (Lightman et al., 2023) and SVAMP (Patel et al., 2021), and two *commonsense reasoning* datasets, ARC-Challenge (Clark et al., 2018), CommonsenseQA (Talmor et al., 2019) for our evaluation.² Examples of prompts are given in Appendix B.

We evaluate our deferral rule on two LLM families: OLMo 2 (OLMo et al., 2025) and Qwen 3

²Although previous works also evaluate on TruthfulQA (Lin et al., 2022), we did not include this dataset as the gold answer is always A in the dataset, see https://huggingface.co/datasets/truthfulqa/truthful_qa.

(Yang et al., 2025), which are open and closed source, respectively. These families provide different model sizes and checkpoints from which we select two contrasting sizes for our experiments.

We use a temperature of 0.7 as recommended to isolate the benefits of our method. As in previous work, to select the margin hyperparameter λ for our new rule, we evaluate a range of values on a small data subset. We finally select $\{0, 0.1, 0.2\}$ for our final evaluation.

4.1 Results Discussion

Results are presented in Table 1. Our conservative confidence bet rule produces results within the expected interval, that is, improving the performance of both building blocks p and q^* and just under the performance level of the target aligned large model p^* , and this holds across all models and tasks. Compared to KAD’s dual approximation rules, our rule with $\lambda = 0.2$ improves the mixture accuracy on the majority of tasks for OLMo 2, with large improvements in the case of the MATH500 task, reaching, on average, higher accuracy than the other baselines.

For Qwen 3, although our rule achieves higher accuracy on mathematical reasoning tasks, it falls short of KAD’s dual approximation rules on commonsense reasoning tasks. This mitigated performance improvement results in comparable average performance scores with our baselines. This is mainly because the performance of the Qwen 3 base model on the selected tasks is close to that of its aligned version: 71.4 compared to 80.2, whereas OLMo 2 has a 37.4 average accuracy-point gap between p and p^* .

5 Conclusion

In this work, we introduced a new probabilistic graphical model for the proxy-based test-time alignment problem. This enables a better understanding of previous work and allows us to derive a new rejection criterion that leverages additional information from the proxy model to assess the confidence of the large base model.

Acknowledgments

This work is supported by the SEMIAMOR (CE23-2023-0005) and InExtenso (ANR-23-IAS1-0004) project grants given by the French National Research Agency (ANR). This work was granted ac-

cess to the HPC resources of IDRIS under the allocation 2024-AD011015801 made by GENCI.

Limitations

As per previous work, hyperparameter selection requires tuning on a separate development set. We show however that even the absence of our margin hyperparameter ($\mu = 0$) can lead to very competitive performances.

References

- Meng Cao, Mehdi Fatemi, Jackie CK Cheung, and Samira Shabanian. 2023. [Systematic rectification of language models via dead-end analysis](#). In *The Eleventh International Conference on Learning Representations*.
- C. Chow. 1970. [On optimum recognition error and reject tradeoff](#). *IEEE Transactions on Information Theory*, 16(1):41–46.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Haikang Deng and Colin Raffel. 2023. [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*, pages 11781–11791, Singapore. Association for Computational Linguistics.
- Tu Anh Dinh and Jan Niehues. 2025. [Are generative models underconfident? better quality estimation with boosted model probability](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3364–3382, Suzhou, China. Association for Computational Linguistics.
- Gonçalo Faria and Noah A. Smith. 2025. [Sample, don’t search: Rethinking test-time alignment for language models](#). *Preprint*, arXiv:2504.03790.
- Yu Fei, Yasaman Razeghi, and Sameer Singh. 2025. [Nudging: Inference-time alignment of LLMs via guided decoding](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12702–12739, Vienna, Austria. Association for Computational Linguistics.

- Leo Gao, John Schulman, and Jacob Hilton. 2023. [Scaling laws for reward model overoptimization](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.
- Ayoub Hammal, Pierre Zweigenbaum, and Caio Corro. 2026. [KAD: A framework for proxy-based test-time alignment with knapsack approximation deferral](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3854–3872, Rabat, Morocco. Association for Computational Linguistics.
- Maxim Khanov, Jirayu Burapachee, and Yixuan Li. 2024. [ARGS: Alignment as reward-guided search](#). In *The Twelfth International Conference on Learning Representations*.
- Nathan Lambert, Jacob 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, Christopher Wilhelm, Luca Soldaini, and 4 others. 2025. [Tulu 3: Pushing frontiers in open language model post-training](#). In *Second Conference on Language Modeling*.
- Bolian Li, Yanran Wu, Xinyu Luo, and Ruqi Zhang. 2025. [Reward-shifted speculative sampling is an efficient test-time weak-to-strong aligner](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11468–11478, Suzhou, China. Association for Computational Linguistics.
- Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. [Making language models better reasoners with step-aware verifier](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. [Let’s verify step by step](#). *Preprint*, arXiv:2305.20050.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A. Smith. 2024. [Tuning language models by proxy](#). In *First Conference on Language Modeling*.
- Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2024. [An emulator for fine-tuning large language models using small language models](#). In *The Twelfth International Conference on Learning Representations*.
- Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Seungyeon Kim, Neha Gupta, Aditya Krishna Menon, and Sanjiv Kumar. 2025. [Faster cascades via speculative decoding](#). In *The Thirteenth International Conference on Learning Representations*.
- Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. [2 OLMo 2 furious](#). *Preprint*, arXiv:2501.00656.
- Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. [Are NLP models really able to solve simple math word problems?](#) In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.
- Ahmad Rashid, Ruotian Wu, Julia Grosse, Agustinus Kristiadi, and Pascal Poupart. 2025. [A critical look at tokenwise reward-guided text generation](#). In *Second Conference on Language Modeling*.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sergey Troshin, Vlad Niculae, and Antske Fokkens. 2025. [On the low-rank parametrization of reward models for controlled language generation](#). *Transactions on Machine Learning Research*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

A Proof of Proposition 1

Proof. Let $\alpha \in [0, 1]$ such that the condition in Equation 1 holds. For all $v \in V$, we write:

$$\begin{aligned}\pi(r = 1 \mid \bar{x} = v) &= \frac{p_v + q_v^* \alpha - s_v}{p_v} \\ &= \mu_v.\end{aligned}\quad (2)$$

We must first prove that $\pi(r \mid \bar{x})$ is a valid distribution, that is $\mu_v \in [0, 1], \forall v \in V$.

First, we have

$$\begin{aligned}\frac{p_v + q_v^* \alpha - s_v}{p_v} &\leq 1 \\ \iff p_v + q_v^* \alpha - s_v &\leq p_v \\ \iff \alpha &\leq \frac{s_v}{q_v^*},\end{aligned}$$

which is true based on our assumptions on α . Second, we have

$$\begin{aligned}\frac{p_v + q_v^* \alpha - s_v}{p_v} &\geq 0 \\ \iff p_v + q_v^* \alpha - s_v &\geq 0 \\ \iff \alpha &\geq \frac{s_v - p_v}{q_v^*},\end{aligned}$$

which is also true based on our assumptions on α .

We can now compute the resulting PGM distribution. By definition, we have:

$$\pi(x = v) = p_v(1 - \mu_v) + q_v^* \sum_w p_w \mu_w.$$

Developing the rejection probability μ_v as per Equation 2, we obtain:

$$\begin{aligned}&= p_v \left(1 - \frac{p_v + q_v^* \alpha - s_v}{p_v} \right) \\ &\quad + q_v^* \sum_w p_w \frac{p_w + q_w^* \alpha - s_w}{p_w} \\ &= p_v - p_v - q_v^* \alpha + s_v \\ &\quad + q_v^* \left(\sum_w p_w + \sum_w q_w^* \alpha - \sum_w s_w \right) \\ &= s_v - q_v^* \alpha + q_v^* (1 + \alpha - 1) \\ &= s_v - q_v^* \alpha + q_v^* \alpha \\ &= s_v,\end{aligned}$$

which ends the proof. \square

B Dataset examples

Following previous work, we rely on regular expressions to extract the final response from the LLM generation. In the case of GSM8K and SVAMP, we extract the last number in the model's response. In the case of MATH500, we extract the last `\boxed{\}` number in the model's response. In the multiple-choice question tasks, we extract the last capital letter from A to J in the response.

The models are evaluated on the question prompts alone, with no additional few-shot examples. For each task, we use the extracted answer's accuracy to evaluate the model's performance. We also evaluate the generation throughput as the average number of tokens generated by the model each second.

The following system prompt was used in all tasks:

Answer the question by walking through the reasoning step by step.

MATH500. Here is an example of a question/answer from MATH500 dataset:

```
{
  "role": "user",
  "content": "Convert the point $(0,3)$ in rectangular coordinates to polar coordinates. Enter your answer in the form $(r,\theta)$, where $r > 0$ and $0 \le \theta < 2\pi$. \n\nPresent the answer in LaTeX format: \n\n\boxed{Your answer}"
},
{
  "role": "assistant",
  "content": "We have that $r = \sqrt{0^2 + 3^2} = 3$. Also, if we draw the line connecting the origin and $(0,3)$, this line makes an angle of $\frac{\pi}{2}$ with the positive $x$-axis. \n\n[asy]\nunitsize(0.8 cm); \n\ndraw((-0.5,0)--(3.5,0)); \n\ndraw((0,-0.5)--(0,3.5)); \n\ndraw(arc((0,0),3,0,90),red,Arrow(6)); \n\ndot((0,3), red); \n\ndot((0,3), W); \n\ndot((3,0), red); \n[/asy] \n\nTherefore, the polar coordinates are $\boxed{\left( 3, \frac{\pi}{2} \right)}$."
```

The following instruction is appended at the end of the user's query :

\n\nPresent the answer in LaTeX format: \n\n\boxed{{Your answer}}

GSM8K. Here is an example of a question/answer from GSM8K dataset:

```
{
  "role": "user",
  "content": "Janet\u2019s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?"
},
```

```
{
  "role": "assistant",
  "content": "How many eggs does Janet sell? ** Janet
sells 16 - 3 - 4 = <<16-3-4=9>>9 duck eggs a day.
\nHow much does Janet make at the farmers' market?
** She makes 9 * 2 = $<<9*2=18>>18 every day at the
farmer\u2019s market.
\n#### 18"
}
```

ARC. Here is an example of a question/answer from ARC dataset:

```
{
  "role": "user",
  "content": "Choose the correct answer to the
following multiple-choice question.
\n\nQuestion: An astronomer observes that a
planet rotates faster after a meteorite impact.
Which is the most likely effect of this increase
in rotation?\n\nA). Planetary density will
decrease.\nB). Planetary years will become longer.
\nC). Planetary days will become shorter.
\nD). Planetary gravity will become stronger.
\n\nProvide your reasoning about the answer and
finish your answer with the letter corresponding
to the correct option (e.g., A, B, C, or D).\n\n"
},
{
  "role": "assistant",
  "content": "\nAnswer: C\n\n"
}
```

The following prefix is prepended to the user's query

Choose the correct answer to the following multiple-choice question.\n\n

The following suffix is appended to the user's query

\nProvide your reasoning about the answer and finish your answer with the letter corresponding to the correct option (e.g., A, B, C, or D).\n\n