RoBoN: Routed Online Best-of-n for Test-Time Scaling with Multiple LLMs

Jonathan Geuter
School of Engineering and Applied Sciences
Harvard University
jonathan.geuter@gmx.de

Gregor Kornhardt
Department of Mathematics
Technische Universität Berlin
g.kornhardt@gmail.com

Abstract

Best-of-n is a widely used test-time scaling approach for LLM inference. Yet despite evidence that LLMs exhibit complementary strengths across tasks, traditionally best-of-n relies on a single model to generate responses. We propose RoBoN (*Routed Online Best-of-n*), a *sequential multi-LLM* alternative to the prevailing *single-model* best-of-n. Given a suite of models $\{m_i\}_{i=1}^M$, RoBoN sequentially routes generations one-by-one across models, based on scores computed using a reward model and an agreement signal on the predicted responses. This online routing requires no additional training, keeps compute parity, and works with any plug-in reward model. Across three math benchmarks (MATH500, OlympiadBench, MinervaMath), RoBoN consistently outperforms standard best-of-n applied to each individual model, with gains up to 5% absolute accuracy, and also improves over a uniform multi-model portfolio baseline. Our results indicate that diversity across models can be exploited *sequentially* at inference to realize better best-of-n performance than any constituent model alone, providing a simple, training-free path to test-time scaling with multiple LLMs.

1 Introduction

Large language models (LLMs) have demonstrated remarkable performance across diverse language tasks, and scaling model and data size has long been a reliable way to enhance capabilities [Kaplan et al., 2020, Reid et al., 2024, OpenAI et al., 2024]. However, in recent years there is growing evidence that performance improvements from scaling training compute decelerate [Hernandez et al., 2022, Muennighoff et al., 2023]; furthermore, ever-larger models incur substantial computational and economic costs. This motivated a shift towards *test-time scaling* strategies that spend more inference compute instead of training compute [Snell et al., 2024, Zhang et al., 2025, Brown et al., 2025]. Different techniques to scale inference compute exist, such as making models “think" longer [Muennighoff et al., 2025], *best-of-n* (BoN) [Gao et al., 2023, Mroueh and Nitsure, 2025, Beirami et al., 2025], or *soft BoN* [Verdun et al., 2025, Geuter et al., 2025].

It is well-known that different LLMs can exhibit complementary strengths [Chen et al., 2025], which motives *ensembling* models in some way. Prior work on ensembling multiple LLMs—e.g., ranking-and-fusion approaches—shows that combining models can outperform any individual model [Jiang et al., 2023], and recent surveys emphasize the opportunity to exploit model diversity at inference [Zhang et al., 2025]. Yet best-of-n is typically employed with a single model, leaving cross-model diversity untapped.

Contributions. We propose RoBoN (*Routed Online Best-of-n*), a *sequential, multi-LLM* alternative to single-model BoN. Given a suite $\{m_i\}_{i=1}^M$ of models, and a per-prompt budget n, RoBoN routes

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Foundations of Reasoning in Language Models.

one generation at a time across models. At each step it evaluates each model's current head candidate with a scorer that combines a plug-in reward model and an agreement signal over predicted answers, commits the best candidate, and recycles the unchosen heads to the next step. This online policy requires no additional training, preserves compute parity (exactly n samples generated in total), and can be layered on top of existing acceleration stacks (e.g., vLLM [Kwon et al., 2023]). On math reasoning benchmarks (MATH500 [Lightman et al., 2024], OlympiadBench [He et al., 2024], MinervaMath [Lewkowycz et al., 2022]), RoBoN consistently improves over BoN with each individual model and over a uniform multi-model portfolio, with gains up to 5% absolute accuracy.

2 Background

Notation. Let $\mathcal{V}$ denote a (finite) vocabulary. Let $\mathcal{X} = \bigcup_{n\in\mathbb{N}} \prod_{i=1}^{n} \mathcal{V}$ be the (countable) space of input prompts, and $\mathcal{Y} = \bigcup_{n\in\mathbb{N}} \prod_{i=1}^{n} \mathcal{V}$ the (countable) space of responses; while $\mathcal{X}$ and $\mathcal{Y}$ are identical as sets, we distinguish them for clarity. We are given a set of models $\{m_1, ..., m_M\}$. For a response generated by model m_i, we will typically write a superscript y^i to denote the model that generated the response. If model m_i generates multiple responses, we will distinguish them with subscripts, e.g. y^i_j, $j = 1, ..., n$. Let $\Delta(\mathcal{Y})$ denote the set of probability measures over $\mathcal{Y}$. For $x \in \mathcal{X}$ and model m, let $\pi_m(y \mid x) \in \Delta(\mathcal{Y})$ be the model's conditional distribution. We assume access to a reward model $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that assigns a (scalar) outcome score $r(x, y)$ to a response y for prompt x. In math tasks we additionally extract a normalized answer $a(y)$ (typically, this will appear inside a `boxed` environment in the answer string y) used for agreement statistics; see section 3 for details. We write $[M] = \{1, \ldots, M\}$ for model indices and use n for the per-prompt test-time sample budget.

Test-time scaling and BoN. Test-time scaling increases inference-time compute per prompt to improve quality. Broadly, test-time scaling methods can be divided into *parallel* and *sequential* approaches, where the former scales test-time computation in parallel with independently generated outputs, while the latter scales inference via computations that sequentially depend on previous computations. Various test-time scaling methods have been explored. Sequential approaches include making models 'think' longer by appending *think tokens* [Muennighoff et al., 2025], or self-correction [Qu et al., 2024]. Common parallel approaches include majority voting [Wang et al., 2023], where n responses are generated in parallel, and the most frequent response is picked; or best-of-n [Beirami et al., 2025], which instead picks one of the n responses using an auxiliary reward model. Concretely, given a single model m and reward model r, *best-of-n* (BoN) draws $y_{1:n} \overset{\text{i.i.d.}}{\sim} \pi_m(\cdot \mid x)$ and outputs

$$y^\star \;=\; \arg\max_{j\in[n]} \, r(x, y_j),$$

which can be seen as a hard-max selection rule. *Soft BoN* instead defines weights

$$w_j \;\propto\; \exp\{\beta\, r(x, y_j)\}, \qquad \beta > 0,$$

and samples $y_j \sim w_j$. As $\beta \to \infty$, soft BoN recovers hard BoN; smaller β interpolates toward averaging and can reduce pathologies of extreme selection [Verdun et al., 2025]. Recent work studies statistical and computational aspects of BoN-style policies, including conditions under which they are (nearly) optimal at fixed inference budgets [Beirami et al., 2025, Huang et al., 2025a, Geuter et al., 2025].

Model ensembling and routing. Multiple LLMs often exhibit complementary strengths, motivating *ensembling* and *routing* across models. Output-level ensembling with learned rankers and fusion can outperform any constituent model [Jiang et al., 2023], and mixture-of-agents (MoA) style approaches aggregate diverse model outputs for additional gains (and debates about when/how to mix) [Jiang et al., 2023, Wang et al., 2025].

Routing has been explored at *training time* via Mixture-of-Experts (MoE), where a learned router activates a small subset of experts per token/layer (where *experts* replace the usual feed-forward heads in attention blocks, and typically specialize in certain parts of the domain) to scale performance while controlling compute and active parameter count [Shazeer et al., 2017, Zhou et al., 2022, Cai et al., 2025].

At *inference time*, *model-level routing* selects among whole LLMs per input, as it is often the case that different LLMs excel at different tasks [Chen et al., 2024, Shnitzer et al., 2024, Huang et al., 2025b]. This usually involves training a parametric router which learns to select the best model for a given prompt.

Unlike parametric routers that must be trained (either within MoE or as separate model selectors), our approach is *training-free* and operates *online*: we sequentially route a fixed budget of generations across a *portfolio* of off-the-shelf LLMs using only (i) a plug-in reward model and (ii) an agreement signal over predicted answers. This preserves compute parity with single-model BoN (exactly n samples per prompt) while exploiting cross-model diversity without learning a router. However, one key difference to vanilla best-of-n is that our approach falls under the *sequential* test-time scaling umbrella, whereas regular best-of-n is *parallel* in nature.

3 Routed Online Best-of-n

Overview. Given a portfolio of models $\{m_i\}_{i=1}^M$, a prompt x, a per-prompt budget n, and a reward model $r(x, y)$, ROBON allocates the n generations *sequentially*. At each step, it evaluates the *current head* candidate from every model with the reward model (the next unseen sample if that model was chosen last round; otherwise the previously drawn sample is *reused*), computes a marginal score for appending each candidate-response pair to the selected set of response-reward pairs S, and greedily commits the best one. This "recycle-unchosen-heads" scheduling ensures exact *compute parity*: the procedure draws exactly n responses in total. However, we only iterate $n - M + 1$ times, such that the final set of candidate responses only contains $n - M + 1$ responses. This corresponds to n generated responses, as in the last iteration, $M - 1$ responses are discarded. This is just a technicality to make for a fair comparison to regular best of n (one could use $n + M - 1$ in place of n to recover n responses in the final set S). Once the final set of response-reward pairs S is generated, the output is chosen by best of n on the set S. For details, see Algorithm 1.

Scoring rule. Let $S = \{(y_\ell, r_\ell)\}_{\ell=1}^s$ denote the multiset of currently selected candidates with rewards $r_\ell = r(x, y_\ell)$. We define *reward weights* via the temperature-scaled softmax:

$$w_\ell \;=\; \frac{\exp\{\beta r_\ell\}}{\sum_{j=1}^s \exp\{\beta r_j\}} \quad \text{for } \ell = 1, \dots, s.$$

We also extract a canonicalized answer string $a_\ell \equiv a(y_\ell)$ (e.g., boxed final value for math). Let

$$\mathrm{agree}_\ell(S) \;=\; \frac{1}{s}\sum_{j=1}^{s} \mathbf{1}\{a_j = a_\ell\} \tag{1}$$

denote the (empirical) agreement of y_ℓ with the current set S. The *agreement-weighted soft score* used to rank candidates is

$$\mathrm{Score}(S; \alpha, \beta) \;=\; \sum_{\ell=1}^{s} w_\ell \cdot \Big(\alpha\, r_\ell + (1{-}\alpha)\, \mathrm{agree}_\ell(S)\Big), \qquad \alpha \in [0, 1].$$

At each iteration of Algorithm 1, for each model i we form the tentative set $S \cup \{(y^i, r^i)\}$ by adding its current head and compute the scores

$$\hat{\Delta}_i \;=\; \mathrm{Score}\big(S \cup \{(y^i, r^i)\}; \alpha, \beta\big).$$

Intuitively, these scores are a measure of how much value a response (y^i, r^i) adds to the existing set S. They can be thought of as an interpolation between best-of-n and majority voting, where the reward term drives selection towards high-reward responses, whereas the agreement term acts as a majority vote incentive, prioritizing responses that occur often in S, which can mitigate reward hacking [Skalse et al., 2025], as we show in Section 4. A seemingly simpler approach would be to compute $\mathrm{Score}(S; \alpha, \beta) = \alpha\, r_\ell + (1{-}\alpha)\, \mathrm{agree}_\ell(S)$ instead. However, surprisingly we found this to not work well, and the above score to work much better. This might be since it computes a "marginal" score over the existing set of responses, instead of a score that does not use information from S. We note that our algorithm seems to be quite robust to different values of α, as long as $\alpha < 1$; see Appendix A.1 for an ablation.

Agreement implementation details. Equation (1) uses a normalized histogram over canonical answers in S. In practice, we compute the agreement term by extracting the answer from the response, applying canonical normalizations (such as removing whitespaces and turning everything into lower case), and then comparing the resulting strings.

Algorithm 1 ROUTED ONLINE BEST-OF-n (ROBON)

Require: Models $m_1, \ldots, m_M$; prompt x; reward model r; budget n; temperature $\beta > 0$, $\alpha > 0$.

$S \leftarrow \emptyset$ ▷ multiset of selected pairs (r, a)
$c_i \leftarrow 1$ for all $i \in [M]$ ▷ head pointer per model
if $n = 1$ **then**
  $i^\star \leftarrow \text{randomChoice}(\{1, ..., M\})$
  $y_1^{i^\star} \leftarrow m_{i^\star}(\cdot \mid x)$
  return $y_1^{i^\star}$
for $t = 1$ **to** $n - M + 1$ **do**
  for $i = 1$ **to** M **do**
    $y_{c_i}^i \leftarrow m_i(\cdot \mid x)$ ▷ generate new response only if c_i increased in previous iteration
    $r_{c_i}^i \leftarrow r(x, y_{c_i}^i)$
    $\hat{\Delta}_i \leftarrow \text{SCORE}(S \cup (y_{c_i}^i, r_{c_i}^i), \alpha, \beta)$
  $i^* \leftarrow \arg\max_i \hat{\Delta}_i$
  $S \leftarrow S \cup (y_{c_{i^*}}^{i^*}, r_{c_{i^*}}^{i^*})$
  $c_{i^*} \leftarrow c_{i^*} + 1$ ▷ increase response counter only for this model
$y \leftarrow \text{BEST-OF-N}(S)$
return y

Algorithm 2 SCORE(S, α, β): agreement-weighted softmax over rewards

Require: Multiset $S = \{(y_\ell, r_\ell)\}_{\ell=1}^s$; weight $\alpha > 0$; temperature $\beta > 0$.

$w \leftarrow \text{softmax}(\beta r), \quad r = (r_1, ..., r_s)$
for $\ell = 1$ **to** s **do**
  $\text{agree}_\ell \leftarrow |\{y \in S : y_\ell = y\}|/s$ ▷ Compute agreement scores
return $\sum_{\ell=1}^s w_\ell \cdot (\alpha r_\ell + (1 - \alpha)\text{agree}_\ell)$

Special cases and limits. When M=1 and α=1, ROBON recovers standard soft best-of-n for a single model. For α=0 the policy becomes agreement-driven (majority-style) selection. For large β and moderate α – the setting we use in our experiments – the policy approximates hard best-of-n with an agreement modified reward over a suite of models, which is particularly effective when models exhibit *complementary strengths*.

4 Experiments

Datasets. We evaluate on three math datsets: MATH500 [Lightman et al., 2024], OlympiadBench, specifically the `OE_TO_maths_en_COMP` split [He et al., 2024], and Minerva-Math [Lewkowycz et al., 2022]. For each dataset, we report the accuracy averaged over the entire dataset (Figure 1 and Tables 1–3), as well as the 1-sigma confidence intervals in Tables 1–3.

Models (M=4). We use the following four models: Qwen2.5-Math-7B-Instruct, DeepSeek-Coder-6.7B-Instruct, Llama-3.1-8B-Instruct, Qwen2.5-Coder-7B-Instruct. We did not perform any search over models, and stuck with our initial suite of models throughout. This suggests ROBON can work out-of-the-box on suites of models of comparable size, and with potentially even larger benefits for different suites. We use Skywork/Skywork-Reward-V2-Llama-3.1-8B as the reward model.

Implementation. We implement all models with vLLM [Kwon et al., 2023]. All experiments ran on a single H100 GPU. The implementation is available at `https://github.com/j-geuter/RoBoN`, where we also provide the full dataset of all generated responses by all four models on all three datasets, with corresponding rewards and normalized rewards.

Hyperparameters. We set $\alpha = 0.4$ (see Appendix A.1 for an ablation). We found that larger values of β work better, so we use $\beta = 1e5$ in our experiments, which essentially recovers picking an $\arg\max$. We generate responses with `temperature` $= 1.0$ and `top_p` $= 0.95$.

Baselines. We compare against the following baselines:
(a) **single-model BoN:** We run regular best-of-n on each separate model from the model pool.

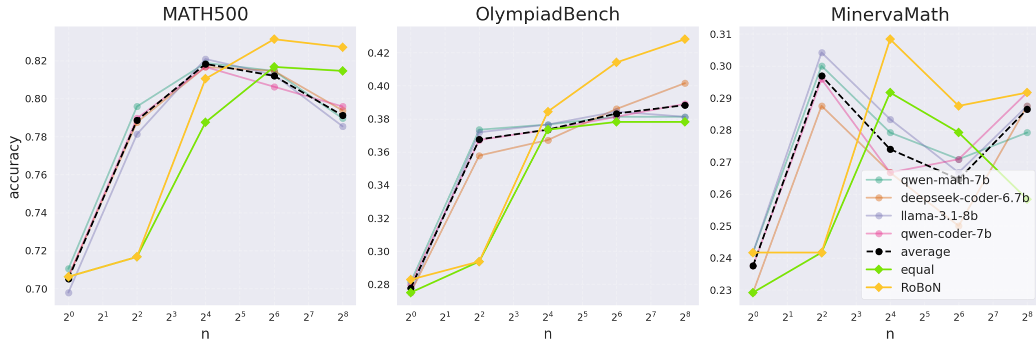

Figure 1: Average accuracies across datasets and methods. For larger n, RoBoN significantly outperforms all baselines. The degrading performance of all methods on MATH500 and MinervaMath as n increases is likely due to reward hacking [Skalse et al., 2025].

Table 1: MATH500. We report average accuracy with 1-sigma confidence intervals.

Method	$n = 16$	$n = 64$	$n = 256$
BoN (Qwen-Math-7B)	0.821 ± 0.024	0.815 ± 0.024	0.790 ± 0.022
BoN (DeepSeek-Coder-6.7B)	0.817 ± 0.024	0.815 ± 0.023	0.794 ± 0.019
BoN (Llama-3.1-8B)	$\mathbf{0.821} \pm 0.024$	0.813 ± 0.024	0.785 ± 0.022
BoN (Qwen-Coder-7B)	0.817 ± 0.027	0.806 ± 0.023	0.796 ± 0.021
Average (across models)	0.818 ± 0.025	0.812 ± 0.023	0.791 ± 0.021
Equal portfolio	0.788 ± 0.022	0.817 ± 0.025	0.815 ± 0.019
RoBoN (ours)	0.810 ± 0.020	$\mathbf{0.831} \pm 0.019$	$\mathbf{0.827} \pm 0.018$

(b) **average:** We average the accuracies of the individual best-of-n strategies from (a).
(c) **equal:** We run best-of-n, where we assign each model an equal share of $n/M = n/4$ samples (for $n = 1$, we pick a model at random).

Reward Normalization. For each model, we normalize rewards by their empirical CDF. Concretely, starting from a large pre-computed corpus of responses and rewards across datasets, we rank the rewards and map them uniformly onto values in $[0, 1]$. This is necessary as otherwise, rewards would not be comparable across models. Indeed, without reward normalization, the performance of RoBoN degrades to that of the average of the individual best-of-n strategies. Note that while this normalization requires a pre-computed dataset of responses and rewards, once this dataset has been created and the empirical CDFs have been estimated, a parametric map from raw to normalized rewards that can be applied to new samples can easily be constructed.

Compute, Memory, and Runtime. In terms of FLOPs, RoBoN is asymptotically exactly en par with standard best-of-n with a single model. The memory requirements grow linearly with the number of models M. In terms of runtime, in the worst-case, RoBoN suffers an additional factor of n compared to standard best-of-n with a single model, since RoBoN generates sequentially instead of in parallel. However, this assumes sufficient memory, and if memory is limited, the runtime of best-of-n can be the same as that of RoBoN. Furthermore, as can been seen from Figure 1, RoBoN achieves accuracies that best-of-n on any single model cannot achieve, no matter how large n is – in fact, the accuracy of best-of-n often starts deteriorating as n increases beyond a certain point. Hence, even under identical runtime budgets, best-of-n will not achieve the same accuracy as RoBoN. This is the reason we decided not to include an explicit runtime comparison to regular best-of-n: In the regime where RoBoN outperforms best-of-n, a runtime comparison is not possible, as best-of-n is not able to achieve comparable performance according to our experiments.

4.1 Performance of RoBoN on Reasoning Tasks

In Figure 1, we report the average accuracies of all baselines and compare them to RoBoN. We see that for $n \geq 16$, RoBoN outperforms all baselines on all datasets by up to 5% absolute

Table 2: OlympiadBench. We report average accuracy with 1-sigma confidence intervals.

Method	$n = 16$	$n = 64$	$n = 256$
BoN (Qwen-Math-7B)	0.377 ± 0.037	0.381 ± 0.039	0.381 ± 0.038
BoN (DeepSeek-Coder-6.7B)	0.367 ± 0.036	0.386 ± 0.039	0.402 ± 0.038
BoN (Llama-3.1-8B)	0.377 ± 0.038	0.384 ± 0.038	0.381 ± 0.037
BoN (Qwen-Coder-7B)	0.373 ± 0.035	0.381 ± 0.038	0.389 ± 0.039
Average (across models)	0.373 ± 0.037	0.383 ± 0.039	0.388 ± 0.038
Equal portfolio	0.373 ± 0.038	0.378 ± 0.036	0.378 ± 0.036
RoBoN (ours)	$\mathbf{0.384} \pm 0.037$	$\mathbf{0.414} \pm 0.039$	$\mathbf{0.428} \pm 0.038$

Table 3: MinervaMath. We report average accuracy with 1-sigma confidence intervals.

Method	$n = 16$	$n = 64$	$n = 256$
BoN (Qwen-Math-7B)	0.279 ± 0.053	0.271 ± 0.058	0.279 ± 0.046
BoN (DeepSeek-Coder-6.7B)	0.267 ± 0.042	0.250 ± 0.044	0.288 ± 0.041
BoN (Llama-3.1-8B)	0.283 ± 0.039	0.267 ± 0.052	0.288 ± 0.044
BoN (Qwen-Coder-7B)	0.267 ± 0.043	0.271 ± 0.047	$\mathbf{0.292} \pm 0.046$
Average (across models)	0.274 ± 0.044	0.265 ± 0.050	0.286 ± 0.044
Equal portfolio	0.292 ± 0.057	0.279 ± 0.049	0.258 ± 0.045
RoBoN (ours)	$\mathbf{0.308} \pm 0.063$	$\mathbf{0.288} \pm 0.051$	$\mathbf{0.292} \pm 0.051$

accuracy (except on MATH500 for $n = 16$). Furthermore, the performance of all baselines degrades significantly on MATH500 and MinervaMath due to reward hacking. While RoBoN also suffers from reward hacking, the effect is significantly diminished. The results also show that RoBoN achieves accuracies that best-of-n with any single model fails to achieve for *any* n. Furthermore, it is interesting to note that RoBoN does not perform well for $n = 4$, in which case a single response from one of the four models is selected based on the scores. We do not report confidence intervals in Figure 1 for visualization purposes, but we report them in Tables 1–3. Appendix A.1 contains an ablation study over α, as well as details on how often each of the four models is selected by RoBoN in practice.

5 Discussion and Limitations

We presented RoBoN, a sequential multi-LLM version of best-of-n sampling for test-time scaling, which adaptively routes generations to models based on scores computed from rewards and agreement signals of responses. In experiments on reasoning datasets, we show that RoBoN significantly outperforms best-of-n baselines of all individual models, as well as a simple portfolio baseline which assigns n samples equally across models. RoBoN is training-free and can be used with any plug-in reward model and across any suite of (comparable) LLMs. However, while sequential test-time scaling methods are often considered advantageous over parallel ones [Muennighoff et al., 2025], an obvious drawback of RoBoN over regular best-of-n is runtime cost, if best-of-n is implemented in parallel. Yet, RoBoN achieves accuracies standard best-of-n fails to achieve independent of runtime budget. Furthermore, the nature of the agreement term is such that RoBoN in its current form is only applicable to tasks where it can be immediately verified whether two answers are identical, as is often the case on reasoning datasets. For more open-ended tasks, one could potentially replace this string-based comparison with embedding similarities.

In future work, we plan to extend RoBoN to other domains, such as coding, verify its benefits on different suites of models, and develop theoretical guarantees. We note that guarantees for the expected accuracy require further assumptions on the reward model, and even then the nature of the RoBoN scoring algorithm makes it difficult to derive reliable guarantees in practice. Furthermore, semi-parallel versions of RoBoN, that could significantly improve runtime complexity, are an interesting direction for future research. Such variants could e.g. compute part of the responses in parallel, then route the remaining responses to the suite of models based on the already computed responses, and compute them in parallel again.

Acknowledgements

JG is supported by a fellowship from the Kempner Institute for the Study of Natural and Artificial Intelligence at Harvard University.

References

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander Nicholas D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. Theoretical guarantees on the best-of-n alignment policy. In *Forty-second International Conference on Machine Learning*, 2025. URL `https://openreview.net/forum?id=u3U8qzFV7w`.

Bradley Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V Le, Christopher Re, and Azalia Mirhoseini. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling, 2025. URL `https://openreview.net/forum?id=0xUEBQV54B`.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A Survey on Mixture of Experts in Large Language Models . *IEEE Transactions on Knowledge & Data Engineering*, 37(07):3896–3915, July 2025. ISSN 1558-2191. doi: 10.1109/TKDE.2025.3554028. URL `https://doi.ieeecomputersociety.org/10.1109/TKDE.2025.3554028`.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=cSimKw5p6R`.

Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S. Yu. Harnessing Multiple Large Language Models: A Survey on LLM Ensemble, 2025. URL `https://arxiv.org/abs/2502.18036`.

Leo Gao, John Schulman, and Jacob Hilton. Scaling Laws for Reward Model Overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/gao23h.html`.

Jonathan Geuter, Youssef Mroueh, and David Alvarez-Melis. Guided Speculative Inference for Efficient Test-Time Alignment of LLMs. In *ES-FoMo III: 3rd Workshop on Efficient Systems for Foundation Models*, 2025. URL `https://openreview.net/forum?id=cRTWN5iwiy`.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL `https://aclanthology.org/2024.acl-long.211/`.

Danny Hernandez, Tom Brown, Tom Conerly, Nova DasSarma, Dawn Drain, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Tom Henighan, Tristan Hume, Scott Johnston, Ben Mann, Chris Olah, Catherine Olsson, Dario Amodei, Nicholas Joseph, Jared Kaplan, and Sam McCandlish. Scaling Laws and Interpretability of Learning from Repeated Data, 2022. URL `https://arxiv.org/abs/2205.10487`.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J Foster. Is Best-of-N the Best of Them? Coverage, Scaling, and Optimality in Inference-Time Alignment. In *Forty-second International Conference on Machine Learning*, 2025a. URL `https://openreview.net/forum?id=QnjfkhrbYK`.

Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. RouterEval: A Comprehensive Benchmark for Routing LLMs to Explore Model-level Scaling Up in LLMs, 2025b. URL `https://arxiv.org/abs/2503.10657`.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, 2020. URL https://arxiv.org/abs/2001.08361.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention, 2023. URL https://arxiv.org/abs/2309.06180.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving Quantitative Reasoning Problems with Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Youssef Mroueh and Apoorva Nitsure. Information Theoretic Guarantees For Policy Alignment In Large Language Models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Uz9J77Riul.

Niklas Muennighoff, Alexander M Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling Data-Constrained Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=j5BuTrEj35.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy

Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 Technical Report, 2024. URL `https://arxiv.org/abs/2303.08774`.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=DRC9pZwBwR`.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, James Molloy, Jilin Chen, Michael Isard, Paul Barham, Tom Hennigan, Ross McIlroy, Melvin Johnson, Johan Schalkwyk, Eli Collins, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Clemens Meyer, Gregory Thornton, Zhen Yang, Henryk Michalewski, Zaheer Abbas, Nathan Schucher, Ankesh Anand, Richard Ives, James Keeling, Karel Lenc, Salem Haykal, Siamak Shakeri, Pranav Shyam, Aakanksha Chowdhery, Roman Ring, Stephen Spencer, Eren Sezener, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *CoRR*, abs/2403.05530, 2024. URL `https://doi.org/10.48550/arXiv.2403.05530`.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=B1ckMDqlg`.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large Language Model Routing with Benchmark Datasets. In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=Zb0ajZ7vAt`.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and Characterizing Reward Hacking, 2025. URL `https://arxiv.org/abs/2209.13085`.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, 2024. URL `https://arxiv.org/abs/2408.03314`.

Claudio M. Verdun, Alex Oesterling, Himabindu Lakkaraju, and Flavio P. Calmon. Soft Best-of-n Sampling for Model Alignment. *arXiv:2505.03156*, 2025.

Junlin Wang, Jue WANG, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-Agents Enhances Large Language Model Capabilities. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=h0ZfDIrj7T`.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=1PL1NIMMrw`.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A Survey on Test-Time Scaling in Large Language Models: What, How, Where, and How Well?, 2025. URL `https://arxiv.org/abs/2503.24235`.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, zhifeng Chen, Quoc V Le, and James Laudon. Mixture-of-Experts with Expert Choice Routing. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 7103–7114. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/2f00ecd787b432c1d36f3de9800728eb-Paper-Conference.pdf`.

A Appendix

A.1 Ablation over α

We run an ablation over different values of α. In Figure 2, we show the average accuracy (over all three datasets) for $\alpha = 0.0, 0.2, 0.4, 0.6, 0.8, 1.0$. We can see that most values of α yield similar results, with $\alpha = 0.4$ having a slight edge. Only $\alpha = 1$, i.e. putting all the weight on the rewards, sees significantly worse performance. This could be attributed to reward hacking [Skalse et al., 2025].

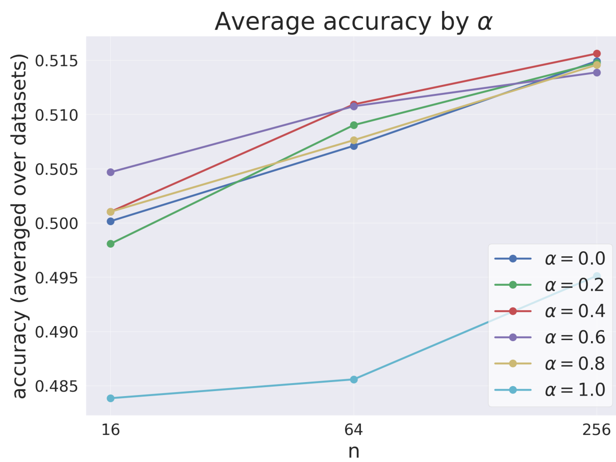

Figure 2: Average accuracy (averaged over all datasets) for RoBoN with different values of α.

A.2 Model Share across n

In Figure 3 we show the average share of models selected by RoBoN across different values of n, averaged over datasets. deepseek-coder-6.7b ranks consistently high with shares between 50-75%. However, the other three models' share seems to help significantly in boosting performance, as RoBoN performs much better than best-of-n on deepseek-coder-6.7b alone, compare Figure 1.

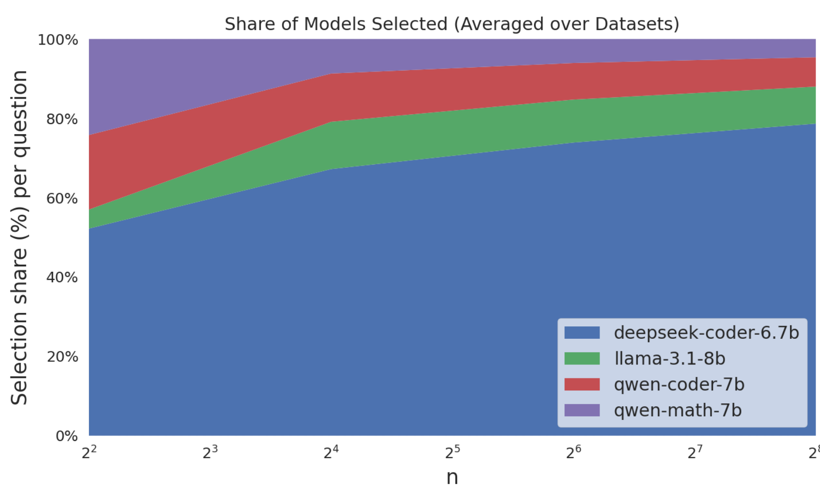

Figure 3: Average share of models selected across different values of n in RoBoN, averaged over datasets. RoBoN selects deepseek-coder-6.7b in the majority of cases; however, RoBoN significantly outperforms this model in terms of accuracy, cmp. Figure 1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: A sequential best-of-n method is introduced, as claimed in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We clearly state limitations and future research, such as developing theory, extending the framework to other types of datasets, and mentioning that runtime is affected by the sequential nature of the method (although sequential test-time methods are also considered to have advantages over parallel methods).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All relevant parameters are included, and all source code is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The full source code including a README with instructions is provided.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details are included.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 1-sigma confidence intervals in Tables 1–3. We do not report them in the plots for visualization purposes.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All compute details are provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper presents general work in machine learning.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper presents a general test-time scaling algorithm for language models. Language models can be misused; however, we don't feel it is necessary to discuss negative impacts of language models in general in this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no high risk of misuse for any of the data we use.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets we used have been cited. All code is our own.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: All documentation for our code is included in the code.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs have not been used in such ways for this paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.