DynScaling: Efficient Verifier-free Inference Scaling via Dynamic and Integrated Sampling

Anonymous ACL submission

Abstract

Inference-time scaling has proven effective in boosting large language model (LLM) performance through increased test-time computation. Yet, its practical application is often hindered by reliance on external verifiers or a lack of optimization for realistic computational constraints. We propose DynScaling, which addresses these limitations through two primary innovations: an integrated parallel-sequential sampling strategy and a bandit-based dynamic budget allocation framework. The integrated sampling strategy unifies parallel and sequential sampling by constructing synthetic sequential reasoning chains from initially independent parallel responses, promoting diverse and coherent reasoning trajectories. The dynamic budget allocation framework formulates the allocation of computational resources as a multi-armed bandit problem, adaptively distributing the inference budget across queries based on the uncertainty of previously sampled responses, thereby maximizing computational efficiency. By combining these components, DynScaling effectively improves LLM performance under practical resource constraints without the need for external verifiers. Experimental results demonstrate that DynScaling consistently surpasses existing verifier-free inference scaling baselines in both task performance and computational cost.

1 Introduction

Inference scaling boosts large language model (LLM) performance through increased test-time computation, offering a cost-effective alternative to retraining (Welleck et al., 2024; Snell et al., 2025; Chen et al., 2024; Wang et al., 2023; Beirami et al., 2024). Early inference scaling efforts primarily aimed for peak performance using substantial computational budgets, frequently leveraging external verifiers like reward models (Brown et al., 2024; Wu et al., 2024; Sardana et al., 2024; Li et al., 2023). Recognizing practical constraints, more recent approaches focus on maximizing performance within a fixed computational budget, acknowledging that this is often more critical for real-world users than achieving theoretical peaks with unlimited computation (Wang et al., 2025; Kang et al., 2025; Chen et al., 2025). In addition, training and deploying verifiers (Cobbe et al., 2021) incur extra cost, and in many practical scenarios, verifiers are either unavailable or unreliable (Gao et al., 2023; Liu et al., 2025b). As a result, recent work has shifted toward more practical settings, emphasizing efficiency under *manageable computational resources* and exploring *verifier-free settings* to improve generalizability (Wang et al., 2025; Kang et al., 2025; Chen et al., 2025).

Verifier-free[1] inference scaling has emerged as a practical and effective strategy for enhancing the performance of LLMs. Recent work highlight that majority voting over responses obtained through a combination of sequential and parallel sampling remains among the most robust and widely applicable approaches across diverse models and tasks (Wang et al., 2025; Chen et al., 2025). Despite the existence of multiple variants, these approaches generally follow a two-dimensional process: generating multiple independent responses in parallel, followed by iterative refinement of each response using the LLM itself in a sequential manner. However, the best practice for applying this strategy, including the proportion of parallel to sequential sampling and the design of the sequential refinement process, can vary across models, tasks, and even individual queries (Snell et al., 2025). These findings give rise to two central questions: (1) How can parallel and sequential inference be **effectively integrated** to optimize performance across diverse models and tasks? (2) How can inference budgets be **dynamically allocated** throughout the sampling process based on intermediate results to improve

[1] Using the LLM itself as the verifier is considered verifier-free.

overall efficiency?

To address the above problems, we propose DynScaling, an efficient, verifier-free inference scaling approach tailored for practical deployment. Our method is grounded in two key innovations: an **integrated parallel–sequential sampling strategy** (§3.2) and a **bandit-inspired dynamic budget allocation framework** (§3.3). First, to better integrate the strengths of both parallel and sequential sampling, we design a strategy that treats the initially independent responses generated in parallel as building blocks for a synthetic sequential reasoning chain. Rather than sampling sequentially from scratch, DynScaling constructs a chain by concatenating these parallel responses to motivate further thinking. To promote diversity in reasoning paths, we also apply random permutations when constructing the chain. This unified view allows DynScaling to benefit from the breadth of parallel sampling and the depth of sequential refinement, while mitigating redundancy. Second, to improve inference scaling efficiency, we formulate budget allocation as a multi-armed bandit (MAB) problem (Robbins, 1952). In this formulation, each query is considered as an arm, and the expected reward associated with pulling an arm corresponds to the uncertainty in the model's current set of responses. Intuitively, queries with higher uncertainty are more likely to benefit from additional computation. We quantify this uncertainty using measures such as the inverse of the majority vote ratio, and adopt an Upper Confidence Bound (UCB) policy (Auer et al., 2002) to balance exploration and exploitation (§3.4). This dynamic allocation mechanism allows DynScaling to focus computation where it is most needed, rather than uniformly distributing budget across all inputs. Together, these components enable DynScaling to achieve strong performance under manageable resources without relying on external verifiers.

We evaluate DynScaling on two reasoning-intensive benchmarks, GPQA and AIME, using both lightweight non-thinking LLM and advanced thinking LLM. Compared to a range of representative verifier-free inference scaling methods, DynScaling consistently achieves higher effectiveness, improved efficiency under practical budget constraints, and greater stability as the inference budget increases. These results underscore the strength of DynScaling 's integrated and process-aware strategies.

In summary, our contributions are as follows:

- We propose an **integrated parallel-sequential sampling strategy** that unifies parallel and sequential sampling by synthesizing multiple independently generated responses into diverse and coherent reasoning trajectories.
- We introduce a **dynamic budget allocation framework** formulated as a multi-armed bandit problem, which adaptively adjusts the inference budget across queries based on the uncertainty of previously sampled responses, thereby improving computational efficiency.
- By combining the above components, we present DynScaling, an **efficient verifier-free inference scaling method**. We empirically demonstrate that DynScaling consistently outperforms representative verifier-free inference scaling baselines in terms of effectiveness, efficiency, and stability.

2 Related Work

Inference Scaling for LLMs. Inference scaling has emerged as a practical strategy to boost the performance of LLMs at test time without requiring retraining. Early work in this area focused on maximizing performance by leveraging extensive computational budgets, often incorporating external verifiers, such as reward models, to guide fine-grained tree search (Feng et al., 2023; Wu et al., 2024) or identify the best response (Brown et al., 2024). While these methods can be effective, their focus on high computational cost regime and dependence on expensive or unreliable verifiers limit their practicality for real-world applications. Recent research has shifted toward more realistic settings that prioritize efficiency under constrained computational budgets, while avoiding the cost and complexity of training or deploying additional verifiers. Following this direction, Wang et al. (2025) benchmark a series of existing methods and identify majority voting as one of the most robust verifier-free strategies. Chen et al. (2025) leverage LLMs themselves as implicit verifiers to develop a structured sequential sampling framework. Other approaches, such as Kang et al. (2025), assume access to token-level logits to develop fine-grained metric to guide generation. Our work builds on this line of research with minimal LLM access by proposing a verifier-free,

Algorithm 1 Bandit-based Dynamic Budget Allocation

Require: Query set $Q = \{q_1, \ldots, q_m\}$, LLM $\mathcal{M}$, Total budget B_{total}, Unit budget B_{unit}

Initialize $B_{\text{used}} \leftarrow 0$
for $q_i \in Q$ **do** ▷ Sample initial responses for each query
  $R_i \leftarrow \texttt{integrated_sampling}(q_i, \mathcal{M}, B_{\text{unit}})$ ▷ Alg. 2
  $B_i \leftarrow B_{\text{unit}}$
  $B_{\text{used}} \leftarrow B_{\text{used}} + B_{\text{unit}}$
end for
while $B_{\text{used}} < B_{\text{total}}$ **do** ▷ Allocate more samples dynamically for prioritized queries
  Compute $a_i \leftarrow \texttt{sampling_priority}(R_i, B_i, B_{\text{used}})$ for each q_i ▷ Alg. 3
  Select subset $\mathcal{S} \subseteq Q$ with top-ranked a_i
  for $q_i \in \mathcal{S}$ **do**
    $R_i \leftarrow R_i \cup \texttt{integrated_sampling}(q_i, \mathcal{M}, B_{\text{unit}})$ ▷ Alg. 2
    $B_i \leftarrow B_i + B_{\text{unit}}$
    $B_{\text{used}} \leftarrow B_{\text{used}} + B_{\text{unit}}$
  end for
end while
for $q_i \in Q$ **do** ▷ Majority voting for the final answer
  $\hat{r}_i \leftarrow \texttt{majority_vote}(R_i)$
end for
return Final responses $\{\hat{r}_i\}_{i=1}^{m}$

model-agnostic method that integrates the strengths of sequential and parallel inference, while dynamically allocating budget across queries to balance effectiveness and efficiency.

Adaptive Inference Budget Allocation. Recent work has explored how to adaptively allocate inference budget either within or across queries for LLMs. Some approaches focus on intra-query adaptation, adjusting internal generation strategies rather than distributing a fixed budget across multiple queries, and are thus orthogonal to our setting. For example, Osca (Zhang et al., 2024) dynamically adjusts the sampling budget for different inference configurations within a query, while DISC (Light et al., 2025) uses a binary decision process to refine selected parts of a previous response. MetaScale (Liu et al., 2025a) employs a UCB-style method to select prompts with diverse personas, enhancing response diversity. In contrast, our work focuses on inter-query budget allocation. Earlier efforts in this direction began with simplified setups, such as those tailored solely for self-consistency (Aggarwal et al., 2023). The most relevant prior efforts include Damani et al. (2025), who train a separate classifier to predict query difficulty based on supervision from a reward model; Manvi et al. (2024), who fine-tune the LLM to act as a generative reward model that determines whether further sampling is needed; and Yu et al. (2025), who learn to control the distribution of response lengths under a constrained reinforcement learning framework. However, all of these methods require additional training and rely on external supervision. Our approach is fully inference-time, model-agnostic, and leverages uncertainty signals from the model's own outputs without any fine-tuning or auxiliary models.

3 Methods

We introduce DynScaling, a verifier-free inference scaling approach for LLMs that achieves more judicious utilization of a fixed inference budget by simultaneously employing parallel and sequential sampling and dynamically allocating inference resources across queries. DynScaling integrates the strengths of parallel and sequential sampling by linking initially independent responses into synthetic thought chains, combining the diversity of parallel sampling with the reasoning depth of sequential prompting. This fusion enhances answer quality at a reduced cost. To further improve efficiency, DynScaling employs a bandit-based dynamic allocation mechanism that adjusts per-query budgets on the fly, guided by signals from earlier responses that balance exploitation and exploration. We begin with a high-level overview of the frame-

Algorithm 2 Integrated Parallel-Sequential Sampling

Require: Query q, Model $\mathcal{M}$, Budget B, Thought length k

$R_{\text{init}} \leftarrow \texttt{sample}(\mathcal{M}, q, B/2)$ ▷ Generate initial responses in parallel
$R_{\text{cond}} \leftarrow [\,]$
for $j = 1$ to $B/2$ **do**
  $T_j \leftarrow \texttt{concat}(\texttt{random_sample}(R_{\text{init}}, k))$ ▷ Combine parallel samples as a sequence
  $r_j \leftarrow \texttt{sample}(\mathcal{M}, q + T_j, 1)$ ▷ Sample more responses based on the integrated sequence
  Append r_j to R_{cond}
end for
return $R_{\text{init}} \cup R_{\text{cond}}$

work, followed by a detailed description of its core algorithmic components.

3.1 Overview

Given a fixed total inference budget, DynScaling aims to maximize LLM output quality across a batch of queries by optimizing how responses are sampled and where the budget is spent. It consists of two key ingredients: (1) an integrated strategy that combines parallel and sequential sampling, and (2) a multi-armed bandit (MAB)-based approach to dynamically prioritize more challenging queries where the model faces uncertainty, so that they can benefit from additional computation. Specifically, DynScaling begins by generating a small set of diverse responses per query using parallel sampling. These are then composed into synthetic thought chains and used as conditional context for additional samples, incorporating sequential reasoning without committing to fixed-depth chains. This integrated sampling approach produces more informative responses with limited cost. Meanwhile, an uncertainty-driven priority mechanism identifies which queries are challenging for the model, dynamically allocating additional sampling budget through a Upper Confidence Bound (UCB)-style acquisition strategy.

3.2 Integrated Parallel-Sequential Sampling

Traditional sampling-based inference methods separately extend parallel sampling (which provides diversity through independent generation) and sequential prompting (which ensures depth through chain-of-thought reasoning). In parallel sampling, the sequences are independent and do not benefit from each other, while sequential methods require multiple steps to reach a fixed depth, making them inefficient. This separation, where parallel sampling focuses on diversity and sequential sampling focuses on depth, often results in suboptimal performance, especially when working within a fixed budget. To effectively leverage the strengths of both strategies, we propose Integrated Parallel-Sequential Sampling (Alg. 2). Given a per-query budget B, we begin by generating $B/2$ independent completions through parallel sampling. Next, we synthesize these responses into intermediate "thought segments" by randomly selecting and concatenating them to create prompt continuations. These synthetic chains provide soft guidance for the remaining $B/2$ samples, encouraging the model to refine or extend promising reasoning from the initial outputs. This hybrid sampling protocol strikes a balance between diversity and depth, enhancing answer quality without requiring costly multi-turn prompting. By integrating parallel and sequential sampling, our approach allows the two to complement and enhance each other, resulting in a more efficient allocation of resources.

3.3 MAB-Based Dynamic Budget Allocation

To further improve budget efficiency across a batch of queries, we introduce a dynamic allocation mechanism (Alg. 1) that adaptively decides which queries should receive higher sampling budget. Inspired by the MAB framework, our method treats each query as a bandit arm and distributes a fixed overall budget by iteratively allocating unit increments to the most promising queries. The process begins with an initial allocation of the unit budget B_{unit} to each query, used to generate a small set of responses via Integrated Parallel-Sequential Sampling. As more samples are collected, the model tracks a priority score for each query, which reflects how informative or uncertain the current responses are. At each step, a subset of high-priority queries is selected to receive additional budget. This allocation continues until the total budget B_{total} is exhausted. This dynamic strategy enables the model

Algorithm 3 UCB-based Sampling Priority

Require: Responses R_i, Budget B_i, Total used budget B_{used}, Exploration ratio c

$u_i \leftarrow \texttt{uncertainty}(R_i)$ ▷ Compute uncertainty as reward

$a_i \leftarrow u_i + c \cdot \sqrt{\frac{\log(B_{\text{used}})}{B_i}}$ ▷ Upper Confidence Bound (UCB)

return a_i

to utilize its generation budget more wisely by investing additional resources in queries that are uncertain or potentially more difficult based on model-specific signals, while avoiding over-sampling on queries for which additional sampling is unlikely to change the final predicted answer. As a result, DynScaling balances coverage across all inputs while adapting sampling effort according to model-perceived difficulty.

3.4 UCB-Style Process-Aware Sampling Priority

Our intuition is that queries eliciting higher uncertainty from an LLM are more likely to benefit from additional sampling. In such cases, generating more responses can increase the chance of uncovering a correct or high-quality answer. Therefore, we use the model's own uncertainty captured from its generated responses as a signal for reward, guiding where to allocate more budget. To operationalize this idea, we adopt a variant of the UCB algorithm (Alg. 3), commonly used in MABs. In our setting, each query acts as an arm, and the model decides which queries should receive more sampling budget based on both their observed uncertainty and how much they have already been sampled. Specifically, we assign each query q_i a priority score:

$$a_i = u_i + c \cdot \sqrt{\frac{\log B_{\text{used}}}{B_i}},$$

where u_i is the reward term representing uncertainty over the current responses for q_i, B_i is the total budget spent so far on query q_i, B_{used} is the total budget consumed across all queries, and c is a tunable hyperparameter that controls the trade-off between exploration and exploitation.

The first term, u_i, encourages *exploitation* by prioritizing queries where the model exhibits high uncertainty and thus stands to gain from further sampling. The second term promotes *exploration* by favoring queries that have been sampled less frequently, ensuring a fair chance of improvement across all queries. We define u_i as the proportion of non-majority predictions (i.e., variation ratio) among current samples:

$$u_i = 1 - \max_{a \in \mathcal{A}_i} \frac{\text{count}(a)}{n_i},$$

where $\mathcal{A}_i$ is the set of unique answers sampled for query q_i, and n_i is the number of samples. This measure reflects the model's uncertainty: it is high when responses are diverse and low when most responses agree. Intuitively, higher u_i indicates that the model has not converged on a dominant answer and may benefit from further sampling. This process-aware priority mechanism adapts to the model's evolving generation behavior. Rather than relying on static notions of difficulty, it tailors budget allocation to the model's own uncertainty signals. As a result, our approach focuses resources on queries that are challenging from the model's perspective, leading to a more effective and efficient use of the inference budget.

4 Experiments

We first describe the experimental setup (§4.1) and then present the main results, comparing DynScaling with verifier-free inference scaling baselines for LLMs under varying budget levels (§4.2). Finally, we provide detailed analyses on different modules (§4.3).

4.1 Experimental Settings

Datasets. We conduct experiments on three datasets widely used to evaluate LLMs, spanning diverse domains and reasoning challenges. **GPQA Diamond** (Rein et al., 2024) comprise PhD-level questions in science.[2] **AIME 24 and AIME 25** are math reasoning datasets.[3] More details are in Appx. §A.1.

LLMs. We conduct experiments using two widely adopted LLMs, with and without the thinking mode introduced by Guo et al. (2025). **Gemini 1.5 Flash** is commonly used in inference-scaling scenarios with no thinking process. **Gemini 2.5**

[2]We refer to this subset simply as GPQA.

[3]We refer to this combined benchmark as AIME.

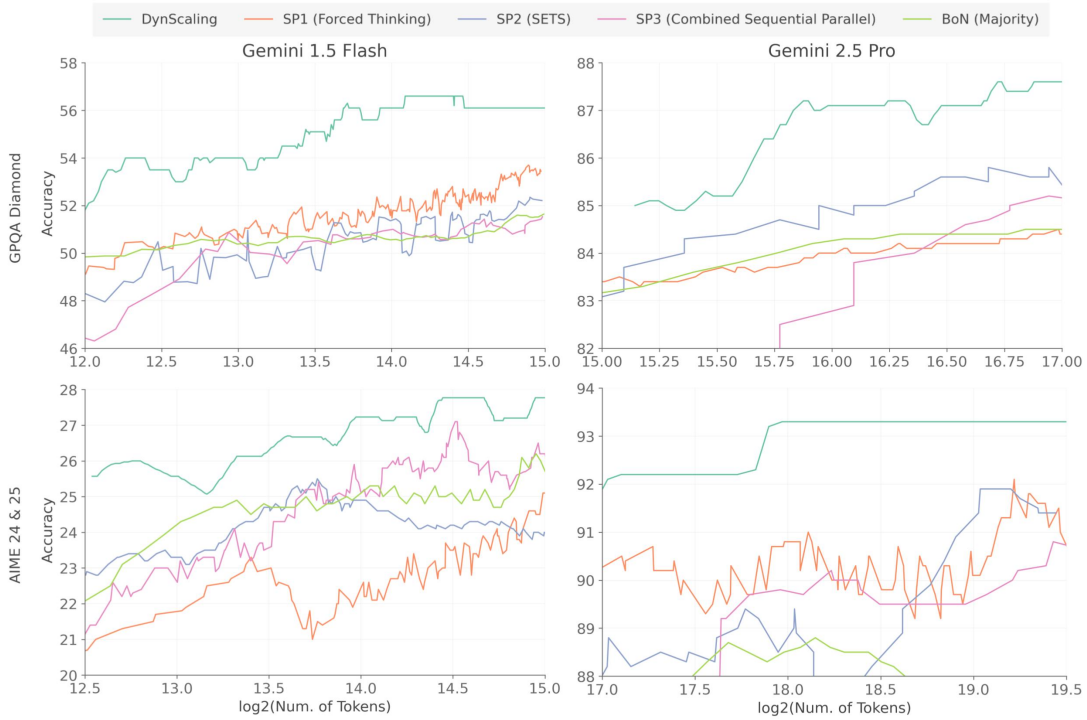

Figure 1: Performance of Gemini 1.5 Flash and Gemini 2.5 Pro on GPQA and AIME across varying inference budgets (measured by the number of output tokens). DynScaling consistently outperforms representative verifier-free inference scaling baselines, demonstrating greater effectiveness, efficiency, and stability across budgets.

Pro is among the most advanced LLMs with thinking mode. More details are in Appx. §A.2.

Metric. Following prior work (Wang et al., 2025; Chen et al., 2025; Muennighoff et al., 2025), we extract the model's predicted answer using pattern matching techniques and evaluate its correctness based on strict equivalence with the ground-truth answer. To assess how performance varies with computational resources, we report model accuracy as a function of the inference budget. However, due to inherent variability, such as changes in sampling depth and breadth when fully utilizing the allocated budget, model performance can fluctuate sharply across nearby budget levels. To reduce this noise and reveal more stable trends, we apply a moving average smoothing technique. Specifically, the reported accuracy at each budget point is computed as the average over a sliding window of adjacent budget values. This helps present a clearer picture of overall performance without being overly sensitive to local variations. All reported results are averaged over three independent runs to account for stochasticity in model generation.

Baselines. We compare DynScaling against a range of *verifier-free inference baselines*, which aim to enhance model performance without relying on external verifiers. Broadly, inference scaling strategies can be categorized along two axes: *sampling breadth*, which involves generating multiple responses in parallel, and *sampling depth*, which expands the reasoning process sequentially. Building on the taxonomy introduced by Wang et al. (2025), we group the baselines into four categories based on how they implement sequential budget expansion. **BoN (Majority)**: This baseline performs *majority voting* over multiple independently sampled responses. It does not involve any sequential expansion. Despite its simplicity, it remains one of the most effective verifier-free inference scaling strategies, as recognized by Wang et al. (2025). **SP1**: This approach involves a *single-step* sequential refinement. Typically, a trigger word or phrase (e.g., "Wait") is appended to the initial response to prompt the model to continue refining its answer, such as *budget forcing* (Muennighoff et al., 2025). **SP2**: This class of methods performs sequential expansion in *two stages*. First, the model generates feedback on its previous answer; second, it refines the previous response based on that feedback (Li et al., 2024). Representative examples include the *self-verification and correction (SETS)* framework proposed by Chen et al. (2025). We adopt the prompt variant from Wang et al. (2025). **SP3**: This approach extends SP2 by incorporating a *third step*: the model evaluates its responses and assigns each a score. The final answer is selected based on a voting mechanism that incorporates these self-assigned scores. We adopt the *combined sequential parallel* framework introduced by Wang et al.

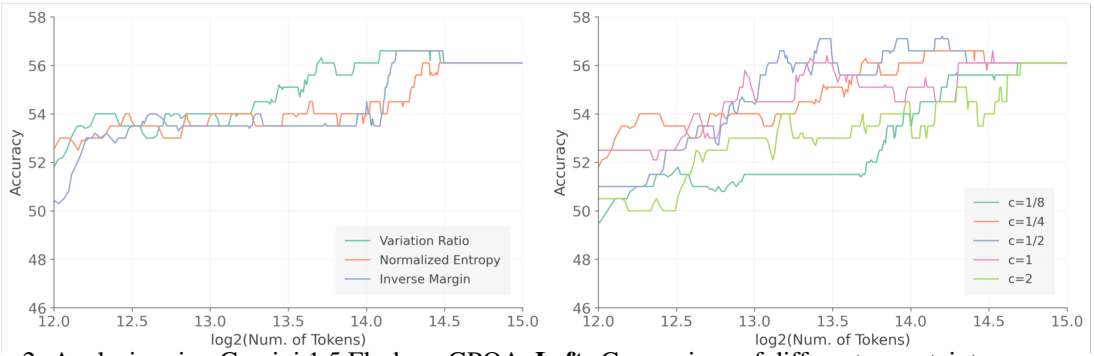

Figure 2: Analysis using Gemini 1.5 Flash on GPQA. **Left:** Comparison of different uncertainty measurements. **Right:** Effect of varying the exploration ratio.

(2025) for implementing SP3.

Implementation Details. We use the number of samples as the budget unit, as it is easier to control, particularly when access to LLMs is limited. Specifically, we set the budget unit B_{unit} to 8 samples. This choice ensures that our integrated parallel-sequential sampling yields $\frac{B_{\text{unit}}}{2} = 4$ predicted answers, which we find to be the minimal effective number for majority voting. The thought length k is set to 4, as increasing k further provides negligible performance gains in our observations. We fix the exploration ratio c to $\frac{1}{4}$ by default, and analyze its impact in §4.3.

4.2 Main Results

Overall Superiority of DynScaling. As shown in Fig. 1, DynScaling consistently outperforms all verifier-free inference scaling baselines across settings, offering superior effectiveness (achieving better performance at the same cost), efficiency (achieving the same performance at a lower cost), and stability (exhibiting smoother trends as the budget increases). These advantages hold across both lightweight models (Gemini 1.5 Flash) and advanced thinking models (Gemini 2.5 Pro), and span diverse tasks including scientific (GPQA) and mathematical (AIME) reasoning.

Efficiency in Low-Budget Regimes. DynScaling demonstrates particularly strong gains under limited inference budgets. For instance, on GPQA with Gemini 1.5 Flash (top-left), it consistently outperforms the best baseline by around 3 accuracy points across most budget levels while maintaining a smoother growth curve. On AIME (bottom-left), it similarly outperforms other methods throughout the budget spectrum, showcasing superior sample efficiency. These results underscore the effectiveness of our dynamic budget allocation strategy under practical, resource-constrained inference budgets.

Scaling Behavior with Thinking Models. With the more capable Gemini 2.5 Pro, DynScaling continues to outperform baselines. On GPQA (top-right), it scales effectively with budget while maintaining a consistent and sizable margin over other approaches. On AIME (bottom-right), although all methods show increased variance at higher budgets, DynScaling avoids the noisy performance drops observed for SP2 and SP3, remaining among the top-performing methods throughout.

Comparison between Sequential and Parallel Strategies. BoN, which scales inference breadth without any sequential refinement, provides steady but modest gains. SP1–SP3 incorporate increasingly complex (structured) sequential prompting but exhibit unstable performance, often fluctuating across budget levels and failing to match the robustness of DynScaling. This highlights the limitations of treating parallel and sequential strategies in isolation, and the advantage of DynScaling's integrated parallel-sequential sampling approach.

4.3 Analyses

We examine how different uncertainty measurements and exploration ratios affect performance as a function of the inference budget. We conduct analyses using the Gemini 1.5 Flash model on the GPQA dataset. Unless otherwise specified, all experiments are run with default settings.

Comparison of Different Uncertainty Measurements As shown in Fig. 2 (left), all three uncertainty estimation methods, including Variation Ratio, Normalized Entropy, and Inverse Margin, exhibit a consistent trend where accuracy improves as more budget is processed, eventually converging to similar values. These results suggest that DynScaling is robust to the choice of uncertainty estimation. Among them, *Variation Ratio* shows a more consistent and gradual increase in accuracy across the entire token range, maintaining

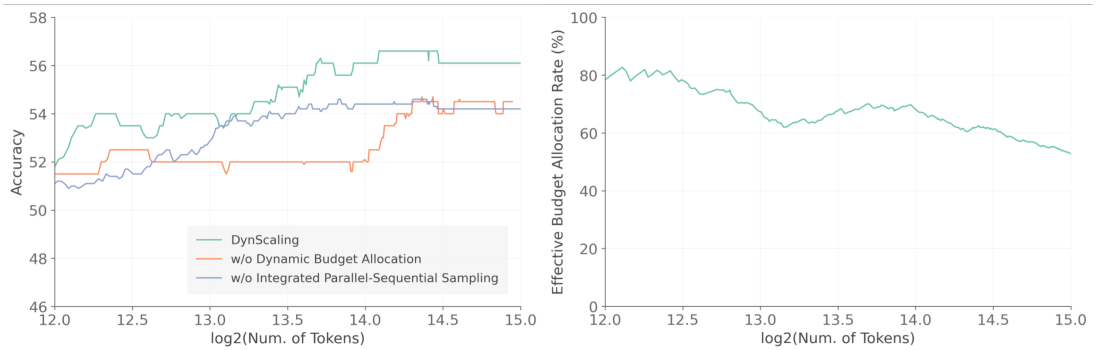

Figure 3: Analysis using Gemini 1.5 Flash on GPQA. **Left:** Ablation study of DYNSCALING components. **Right:** Effective budget allocation toward incorrect queries.

competitive performance throughout. *Normalized Entropy* provides steady accuracy in the lower budget range, followed by an increase at high budget range that brings it on par with Variation Ratio. *Inverse Margin*, while initially underperforming, displays two distinct surges in accuracy at specific budget intervals, eventually catching up with the other methods at higher budget.

Effect of Different Exploration Ratios. We evaluate the impact of varying the exploration ratio c across values of $\frac{1}{8}$, $\frac{1}{4}$, $\frac{1}{2}$, 1, and 2, observing consistent trends in accuracy progression in Fig. 2 (right). Smaller ratios, particularly $c = \frac{1}{8}$ and $c = \frac{1}{4}$, tend to yield higher initial accuracy under limited budget. In contrast, larger values such as $c = \frac{1}{2}$, $c = 1$, and $c = 2$ often catch up or surpass the smaller ratios as the budget increases, benefiting from stronger exploratory behavior in longer processing regimes. While no single value of c is universally optimal, the results suggest that smaller ratios are better suited for low-budget settings, whereas larger ratios offer advantages when more budget is available. In practice, the choice of c can be guided by the expected inference budget.

Ablation Studies of DYNSCALING Components We conduct an ablation analyses to isolate the contributions of key components in DYNSCALING. As shown Fig. 3 (left), removing either *Dynamic Budget Allocation* or *Integrated Parallel-Sequential Sampling* leads to noticeable drops in performance across most budget levels. The complete method consistently outperforms both ablated versions, particularly in the early to mid stages, demonstrating the benefits of effective early exploration and adaptive allocation. The sharp plateau of the *w/o Dynamic Budget Allocation* variant highlights the rigidity introduced by a static budget strategy, while the slower early growth of the *w/o Integrated Parallel-Sequential Sampling* variant reflects the limitations of decoupled sequential and parallel sampling. These trends confirm that both components are essential for the efficiency and accuracy gains achieved by DYNSCALING, especially under tight or moderate budget constraints.

Effective Budget Allocation Toward Incorrect Queries We analyze the *effective budget allocation rate*, defined as the proportion of additional budget allocated to the remaining incorrect queries. As shown in Fig. 3 (right), DYNSCALING maintains a consistently high allocation rate across varying budget levels. When the total budget is low, many incorrect queries are relatively easy to detect, allowing DYNSCALING to allocate up to 80% of the additional budget toward correcting them. As the budget increases, fewer incorrect queries remain, and those that do are harder to identify, leading to a gradual decline in the allocation rate. Nevertheless, even in the most challenging cases within the studied range, the effective allocation rate remains above 50%, highlighting DYNSCALING's robustness in prioritizing error correction even under diminishing returns.

5 Conclusion

We presented DYNSCALING, a practical and verifier-free inference scaling approach that integrates the strengths of both parallel and sequential sampling while dynamically allocating inference budgets. By synthesizing independent responses into a unified reasoning chain and formulating budget allocation as a bandit problem, DYNSCALING effectively balances computational cost and performance. Our experiments on challenging reasoning benchmarks demonstrate that DYNSCALING achieves superior effectiveness, efficiency, and robustness compared to existing verifier-free baselines. These results highlight the potential of process-aware and uncertainty-guided inference strategies for practical deployment of LLMs under constrained resources.

Limitations

While DynScaling is model-agnostic and operates entirely at inference time, it assumes access to a batch of queries to enable cross-query budget reallocation. This requirement may restrict its applicability in latency-sensitive or single-query scenarios, such as real-time interactive systems. Moreover, DynScaling currently relies on simple uncertainty heuristics, which may not capture nuanced ambiguity in model responses and could limit allocation precision in more complex settings. Furthermore, although we focus on reasoning benchmarks in this work, the generalizability of DynScaling to other domains or task types remains to be explored.

References

Pranjal Aggarwal, Aman Madaan, Yiming Yang, et al. 2023. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256.

Ahmad Beirami, Alekh Agarwal, Jonathan Berant, Alexander D'Amour, Jacob Eisenstein, Chirag Nagpal, and Ananda Theertha Suresh. 2024. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and Sercan Ö Arık. 2025. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mehul Damani, Idan Shenfeld, Andi Peng, Andreea Bobu, and Jacob Andreas. 2025. Learning how hard to think: Input-adaptive allocation of LM computation. In *The Thirteenth International Conference on Learning Representations*.

Xidong Feng, Ziyu Wan, Muning Wen, Ying Wen, Weinan Zhang, and Jun Wang. 2023. Alphazero-like tree-search can guide large language model decoding and training. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhewei Kang, Xuandong Zhao, and Dawn Song. 2025. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*.

Yanhong Li, Chenghao Yang, and Allyson Ettinger. 2024. When hindsight is not 20/20: Testing limits on reflective thinking in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3741–3753.

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.

Jonathan Light, Wei Cheng, Wu Yue, Masafumi Oyamada, Mengdi Wang, Santiago Paternain, and Haifeng Chen. 2025. Disc: Dynamic decomposition improves llm inference scaling. *arXiv preprint arXiv:2502.16706*.

Qin Liu, Wenxuan Zhou, Nan Xu, James Y Huang, Fei Wang, Sheng Zhang, Hoifung Poon, and Muhao Chen. 2025a. Metascale: Test-time scaling with evolving meta-thoughts. *arXiv preprint arXiv:2503.13447*.

Runze Liu, Junqi Gao, Jian Zhao, Kaiyan Zhang, Xiu Li, Biqing Qi, Wanli Ouyang, and Bowen Zhou. 2025b. Can 1b llm surpass 405b llm? rethinking compute-optimal test-time scaling. *arXiv preprint arXiv:2502.06703*.

Rohin Manvi, Anikait Singh, and Stefano Ermon. 2024. Adaptive inference-time compute: Llms can predict if they can do better, even mid-generation. *arXiv preprint arXiv:2410.02725*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Herbert Robbins. 1952. Some aspects of the sequential design of experiments.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. 2024. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws. In *International Conference on Machine Learning*, pages 43445–43460. PMLR.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Qwen Team. 2025. Qwen3.

Junlin Wang, Shang Zhu, Jon Saad-Falcon, Ben Athiwaratkun, Qingyang Wu, Jue Wang, Shuaiwen Leon Song, Ce Zhang, Bhuwan Dhingra, and James Zou. 2025. Think deep, think fast: Investigating efficiency of verifier-free inference-time-scaling methods. *arXiv preprint arXiv:2504.14047*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*. Survey Certification.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Zishun Yu, Tengyu Xu, Di Jin, Karthik Abinav Sankararaman, Yun He, Wenxuan Zhou, Zhouhao Zeng, Eryk Helenowski, Chen Zhu, Sinong Wang, et al. 2025. Think smarter not harder: Adaptive reasoning with inference aware optimization. *arXiv preprint arXiv:2501.17974*.

Kexun Zhang, Shang Zhou, Danqing Wang, William Yang Wang, and Lei Li. 2024. Scaling llm inference with optimized sample compute allocation. *arXiv preprint arXiv:2410.22480*.

A Experimental Details

A.1 LLMs

Gemini 1.5 Flash (Team et al., 2024) is a lightweight variant from the Gemini 1.5 family, optimized for efficiency with minimal degradation in quality. Due to its low cost and latency, it is commonly used in inference-scaling scenarios. This model does not support thinking mode and typically produces concise responses. Unless otherwise specified, we set the temperature to 0.6 and the maximum output length to 8,192 tokens.

Gemini 2.5 Pro[4] is among the most advanced LLMs, achieving state-of-the-art performance across a wide range of tasks. It supports thinking mode (Guo et al., 2025; Team, 2025), which enables a structured internal reasoning process before finalizing a response. By default, we enable thinking mode for all responses and set the temperature to 0.6 and the maximum output length to 32,768 tokens.

A.2 Datasets

GPQA Diamond (Rein et al., 2024) is a highly challenging subset of the GPQA benchmark, comprising 198 multiple-choice questions that require PhD-level knowledge in the natural sciences, including biology, chemistry, and physics.

AIME 24[5] **and AIME 25**[6] are derived from the 2024 and 2025 American Invitational Mathematics Examinations, respectively. Each dataset contains 30 problems that test advanced mathematical problem-solving skills, often involving nontrivial, multi-step reasoning. For evaluation, we combine the two sets into a single benchmark. Throughout this paper, we refer to this combined benchmark as AIME.

B Future Work

To further broaden the scope of DynScaling, future work can (1) extend our approach to support fine-grained, token-level or intra-query budget allocation for more nuanced control over inference computation; (2) incorporate richer signals, such as alignment with task-specific priors, to refine the exploitation aspect of the budget allocator; and (3) adapt DynScaling for streaming or interactive settings, where queries arrive incrementally and immediate, per-query decisions are required without access to a query batch.

C Broader Impacts

Our work on DynScaling enhances the efficiency and practicality of LLM inference, potentially broadening access to advanced AI capabilities in domains such as education, healthcare, and scientific research. By eliminating the need for external verifiers, DynScaling lowers deployment barriers and reduces computational costs, making powerful LLMs more accessible to a wider range of users and applications. While DynScaling itself does not introduce new ethical risks, careful evaluation and responsible use remain essential. Future work should explore safeguards and monitoring mechanisms to mitigate potential negative societal impacts.

[4] https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/

[5] https://huggingface.co/datasets/HuggingFaceH4/aime_2024

[6] https://huggingface.co/datasets/opencompass/AIME2025