Internal Value Functions: Leveraging Hidden States for Efficient Test-Time Scaling in Large Reasoning Models.

Anonymous Author(s)
Affiliation
Address
email

Abstract

Large Reasoning Models (LRMs) generate extensive hidden states during inference, which encode rich information about the input context and probabilistically influence future token predictions. We propose Internal Value Functions (IVF), a novel approach that leverages these hidden states to approximate state-value functions, effectively predicting how likely a partial reasoning trajectory will converge to the correct answer without additional inference steps. Unlike traditional Process Reward Models (PRMs) that require separate model evaluations, our method enables efficient implementation of several test-time scaling techniques by extracting predictive signals from intermediate representations computed during the forward pass. Experimental results on challenging reasoning benchmarks demonstrate that IVF achieves comparable or better performance than external PRMs while significantly reducing computational overhead.

1 Introduction and Related Works.

Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities through chain-of-thought (CoT) prompting Wei et al. (2022). Snell et al. (2024) show that Process Reward Models (PRMs) (Lightman et al., 2023) can further improve LLMs performance on complex reasoning tasks Wang et al. (2023); Liu et al. (2024); Zhang et al. (2025b) by guiding the generation process with search. Recently, Large Reasoning Models (LRMs) Guo et al. (2025a) extends LLMs by introducing an explicit thinking phase where models can explore, backtrack, and verify their reasoning before generating final answers. Thus, there is a natural need to develop PRMs for LRMs.

However, traditional PRMs face challenges when applied to LRMs' long-form reasoning. The primary difficulty stems from the non-linear nature of LRM thinking, where initial mistakes can be corrected later in the reasoning process Zou et al. (2025). While recent works have proposed specialized PRMs for LRMs Wang et al. (2025); Zou et al. (2025), these approaches still rely on training separate external models and often struggle with the complexity of long-form reasoning trajectories.

We propose Internal Value Function (IVF), a novel approach that leverages the LRM's own hidden states to approximate its state-value function. Our approach is motivated by recent observations that LRM's hidden states can effectively predict model behavior and exhibit strong calibration properties (Zhang et al., 2025a). We extend Guo et al. (2025b) who use the hidden states to build light-weight outcome reward models (ORM) for non-reasoning LLMs.

Submitted to NeurIPS 2025 Workshop on Efficient Reasoning. Do not distribute.

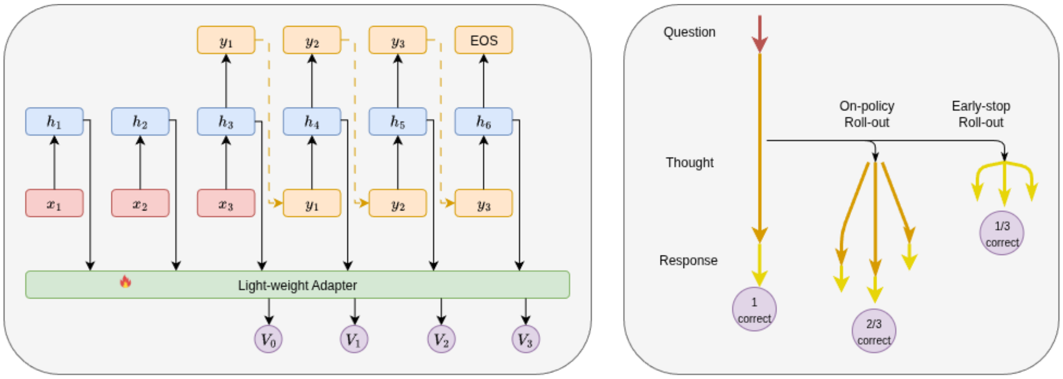

Figure 1: Left: Illustration of our hidden states adapter. V_m is short for $V(y_{0:m}|x)$ in equation 1. The hidden states are simplified - in general, there are multiple layers of hidden states and at each layer they depend on all previous hidden states through causal self-attention. At the top, an LRM auto-regressively decodes tokens y. At the bottom, hidden states h of the unfinished completion are passed to a light-weight adapter that estimates its value, similar to PRMs. Right: The process of generate training labels for each complete trajectory. The first label is the final outcome, which is either 0 or 1. A second label is derived for a partial thought (a.k.a a roll-in) by averaging either multiple On-policy Roll-outs (continuing the thought) or multiple Early-stop Roll-outs (immediately closing the thought and forcing the response). See section 2 for details.

2 Methods

2.1 Problem Formulation

Consider a generator model (policy) π that takes a problem prompt $x = (x_1, ..., x_{m_x})$ and produces a completion $y = (y_1, ..., y_{m_y})$, where each is a sequence of tokens of length m_x and m_y respectively. The completion y is sampled from the conditional distribution $\pi(y|x)$. Unless using greedy sampling (temperature = 0), different calls to the generator with the same prompt will yield different completions, including varying lengths m_y. The policy π encompasses all sampling parameters (temperature, length limit, top-p, top-k, etc.), ensuring a unique conditional distribution.

In reasoning tasks, we assign a binary reward $o(y|x)$ to each completion: 1 for correct answers and 0 otherwise. Our goal is to predict the "potential" of any partial completion $y_{0:m}$ for $0 \leq m \leq m_y$ ($m = 0$ represents the initial state with no completion). Such predictions enable more effective solution search Snell et al. (2024).

2.2 Value Functions

Following Wang et al. (2025), we refer to $y_{0:m}$ as a roll-in and $y_{m+1:m_y}$ as a roll-out. We formulate our PRM using the state-value function Sutton et al. (1998):

$$V(y_{0:m}|x) = \mathbb{E}_{y_{m+1:m_y} \sim \pi(\cdot|x \oplus y_{0:m})}[r(y_{m+1:m_y} \oplus y_{0:m}|x)] \tag{1}$$

where $\oplus$ denotes sequence concatenation. This function represents the expected reward when starting from prompt x with roll-in $y_{0:m}$ and completing the sequence using π. It effectively measures the roll-in's potential to lead to a correct answer.

While it is possible to have different generators for roll-in and roll-out, in this work, we focus on generating a roll-out with the same policy π used for roll-in. The intuition is that a light-weight adapter taking the roll-in hidden states as input is possible if this input is predictive of the roll-out hidden states. We call this **On-policy Roll-out**.

Motivated by recent successes in early-stopping for LRMs (Sui et al.), we also introduce **Early-stop Roll-out**. This variant also uses an on-policy generator but forces the partial completion to exit the thinking phase. Formally, $\pi_{ES}(\cdot|x \oplus y_{0:m})$ matches $\pi(\cdot|x \oplus y_{0:m})$ if x contains the closing tag `"</think>"`, otherwise it samples from $\pi(y|x \oplus y_{0:m} \oplus w)$, where w is a closing phrase:

`"\nOkay, I think I have finished thinking\n</think>\n\n"`.

Table 1: Performance of DVTS across different datasets and models. Results with * are taken from Wang et al. (2025). blue and yellow highlight the best and second-best methods in each column, respectively.

Method	AIME-25	HMMT-25	GPQA-Diamond
RIM	58.5 ± 2.1	36.2 ± 3.8	52.5 ± 7.0
External PRMs:			
Qwen2.5-Math-PRM-7B*	38.9 ± 1.4	24.2 ± 0.2	-
MathShepherd-PRM-7B*	41.9 ± 1.4	23.9 ± 1.4	-
VGS	57.9 ± 2.1	39.1 ± 2.5	52.2 ± 1.4
Our IVFs:			
On-policy Roll-out	58.5 ± 2.2	35.2 ± 3.1	52.9 ± 1.3
Early-stop Roll-out	57.9 ± 1.9	41.2 ± 4.5	54.8 ± 1.4

2.3 Hidden State Representation

Unlike external PRMs that operate on token outputs $y_{0:m}$, we propose leveraging the generator's intermediate hidden states through a lightweight adapter. During generation of y, the model computes hidden states $h_{i,1}, h_{i,2}, ..., h_{i,m_x}$ for prompt x and $h_{i,m_x+1}, h_{i,m_x+2}, ..., h_{i,m_x+m_y} \in \mathbb{R}^d$ for completion y, where $i \in L$ indexes layers in a transformer model with L layers and hidden dimension d.

To manage long generation sequences efficiently, we sample a subset of hidden states. Specifically, following Skean et al. (2025), we select states from a single middle layer (layer 15 for r1-qwen7b) and also sample states at fixed intervals (256 tokens by default). With a slight abuse of notation, we denote the resulting sequence of T sampled hidden states as $h_1, ..., h_T$.

2.4 Adapter Architectures

We extend the simple architecture in Guo et al. (2025b). This model predicts both gating values $\tilde{g}_t$ and rewards r_t:

$$[\tilde{g}_t, r_t] = W h_t + b; \quad g_t = \sigma(\tilde{g}_t); \quad V_{t'} = \frac{\sum_{t=1}^{t'} (g_t \cdot r_t)}{\max(\sum_{t=1}^{t'} g_t, \epsilon)}.$$

While initially used for outcome-based training only, this model readily handles training with intermediate labels. We further add an intermediate self-attention layer to this model, noting that experiments with adding more layers and also a different RNN-based model did not improve this simple baseline. In total, this model has less than 1 million parameters. During training, a binary cross-entropy loss is computed at each time step with a label.

2.5 Training Methodology

Our training procedure follows Equation 1 with several practical considerations. While ideally we would collect ground truth values throughout each sequence using Monte Carlo roll-outs, computational constraints lead us to collect a single ground truth value per training sequence. The data generation process consists of: 1. sampling 8 full completions per prompt (1,000 prompts in total), 2. creating roll-ins by truncating completions at random points during thinking, 3. generating 8 roll-outs per roll-in, and 4. computing value estimates by averaging roll-out rewards. This process yields 8000 training sequences, each with two ground truth values: one at sequence end and one from roll-out averaging. In total, we generate 8000 roll-in samples and 64000 samples for each kind of roll-out. In contrast, our best baseline Wang et al. (2025) generate 2.8 million datapoints to train their 1.5B-parameter external PRM.

Table 2: BoN Diverse Sampling with generator r1-qwen7b across different datasets and PRMs. blue, yellow and green highlight the best, second-best and third-best in each column, respectively.

Method	AIME-25	HMMT-25	GPQA-Diamond
RIM	50.5 ± 3.6	30.2 ± 3.8	53.5 ± 1.9
External PRMs:			
Qwen2.5-Math-PRM-7B	51.7 ± 3.4	34.7 ± 3.9	50.0 ± 1.9
VGS	50.8 ± 2.5	40.7 ± 3.2	49.9 ± 1.5
ReasonFlux-PRM-7B	52.2 ± 3.3	34.8 ± 3.8	49.9 ± 1.8
Our IVFs:			
On-policy Roll-out	54.8 ± 3.4	33.6 ± 4.4	52.1 ± 1.9
Early-stop Roll-out	51.8 ± 3.4	36.1 ± 4.3	54.0 ± 2.0

3 Experimental Results

Generators: We use deepseek-r1-distill-qwen-7b (Guo et al., 2025a) (r1-qwen7b) as the base generator. We use a maximum of 16384 tokens for generation and other sampling parameters as recommended by source, namely temperature = 0.6, top-p = 0.95.

Datasets: We train on 1400 prompts from the DAPO dataset Yu et al. (2025), reserving 400 for validation and test. DAPO, originally designed for reinforcement learning of LRMs, provides high-quality problems with easily verifiable integer answers. We evaluate on 3 benchmarks: Mathematical reasoning: AIME25, HMMT25[1] and Scientific reasoning: GPQA-Diamond (Rein et al., 2024).

Baselines: We compare against various external PRM baselines: MathShepherd-PRM-7B (Wang et al., 2023), Qwen2.5-Math-PRM-7B (Zhang et al., 2025b), ReasonFlux-PRM-7B (Zou et al., 2025), DeepSeek-VM-1.5B (VGS) Wang et al. (2025). For fair comparison and to highlight the impact of process-based training, we also compare with a modified version of the ORM of Guo et al. (2025b) (RIM), which differs with our IVFs only by not having intermediate roll-out labels.

3.1 Diverse Verifier Tree Search

Following Wang et al. (2025), we evaluate the performance with Diverse Verifier Tree Search (DVTS) and report Weighted Majority Voting. We use the same algorithm and hyper-parameters as Wang et al. (2025) that were optimal for their method. In particular, we run 16 independent beam search instances per prompt, each instance uses 8 beams and beam width 2, the total budget is therefore 128. We use bootstrap with 2^{15} resamples to compute the confidence intervals.

Table 1 presents the DVTS results. Our models outperform VGS. In particular, Early-Stop Roll-out IVF achieves the best performance on HMMT25, GPQA and second-best on AIME-25. Strong performance on GPQA demonstrates good generalization to out-of-distribution scientific reasoning, despite training on mathematical problems. These results demonstrate that our IVFs can effectively guide search-based inference, matching or exceeding the performance of external PRMs while using significantly fewer parameters.

3.2 Diverse Sampling with Best-of-N

We also show that our PRM training can improve the ORM training of RIM by report the Divere Sampling with Best-of-N performance Snell et al. (2024). For each prompt, we generate 128 completions and perform bootstrap to evaluate Bo128 performance (Huang et al., 2025).

The results in Table 2 demonstrate process-based training can outperform outcome-based training (RIM) even when the search algorithm only requires the full trajectory score. Second, our light-weight IVFs again achieve comparable or better performance than billion-parameter external PRMs. In particular, Early-stop Roll-out IVF are in the top 3 methods in each dataset.

[1] `https://maa.org/maa-invitational-competitions` and `https://www.hmmt.org`

References

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

Jizhou Guo, Zhaomin Wu, and Philip S Yu. Reward inside the model: A lightweight hidden-state reward model for llm's best-of-n sampling. *arXiv preprint arXiv:2505.12225*, 2025b.

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. *arXiv preprint arXiv:2503.21878*, 2025.

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*, 2023.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

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

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models, 2025. *URL https://arxiv. org/abs/2503.16419*.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Kaiwen Wang, Jin Peng Zhou, Jonathan Chang, Zhaolin Gao, Nathan Kallus, Kianté Brantley, and Wen Sun. Value-guided search for efficient chain-of-thought reasoning. *arXiv preprint arXiv:2505.17373*, 2025.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025a.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025b.

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-prm: Trajectory-aware prms for long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2506.18896*, 2025.