INTROSPECTIVE VERIFICATION: READING STEP CORRECTNESS AT INTERMEDIATE DEPTH

Anonymous authors
Paper under double-blind review

ABSTRACT

Process reward models score the correctness of individual reasoning steps, and these scores drive selection and search in multi step reasoning. The most accurate of these models work through generation, decoding a critique for every step and often calling an interpreter, which makes checking a solution expensive. Whether the same judgment can instead be read directly from the verifier's own hidden states has not been resolved. We use a linear probe that locates the signal at an intermediate layer near seventy percent of the depth, where step correctness is more separable than at the final layer. We call this introspection, since the verdict is decoded from the backbone's own representation rather than generated as text. Once each readout is trained rather than probed, however, the final layer becomes the stronger readout on its own. Our introspective verdict head sums a trained intermediate readout and a trained final readout into a per step reward in a single forward pass, with no generation and no code execution. We adapt the backbone with LoRA on about eight thousand conversations from one released label set, and both training and inference run on a single 32GB V100 at the 1.5B and the 7B scale. At 1.5B the fusion hyperparameters are tuned only on MATH and applied unchanged to three domains never touched during tuning, which shows cross-domain generalization. On ProcessBench the 1.5B head reaches 58.1 mean F1, above most trained process reward models including ones at 7B and 14B scale, at about one quarter the token cost of a single generative pass. Fused with a 7B generative verifier it raises mean F1 from 75.2 to 77.4, without lowering any subset. Code is available at `https://anonymous.4open.science/r/LLM_CALM-48B7`.

1 INTRODUCTION

Reliable multi step reasoning depends on the ability to tell which step first goes wrong, and process reward models supply this signal by scoring each step rather than only the final answer (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024). Such scores drive selection and search at test time and shape rewards during training (Wang et al., 2023; Snell et al., 2024). The labels that train these models are costly. They come from human annotation (Lightman et al., 2024), from Monte Carlo rollouts that estimate whether a step still leads to the correct answer (Wang et al., 2024), or from consensus filtering between rollouts and a language model judge (Zhang et al., 2025b). ProcessBench made step level accuracy directly comparable across methods (Zheng et al., 2025a), and a recent survey maps the fast growth of the area (Zheng et al., 2025b).

Two families of process reward models dominate, classification models and generative models, and each has a clear limitation. Classification models attach a scalar head to the final layer representation and score a step in one forward pass. They are cheap, but they score below stronger verifiers, in particular on proof heavy problems where the answer is not a single number (Zheng et al., 2025a; Zhang et al., 2025b), which has motivated label efficient variants that still read from the final layer (Yuan et al., 2025; Sun et al., 2025). Generative models close the accuracy gap by writing a critique of each step, and some also generate and run code before a verdict (Zhang et al., 2025a; Zhao et al., 2026; Xiong et al., 2025; Khalifa et al., 2026). This accuracy is expensive, because it turns one verdict into a long decoded critique, an interpreter call, and often several samples for majority voting (Snell et al., 2024). Adaptive routing lowers the average cost but keeps generation for the hard steps (Zhong et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

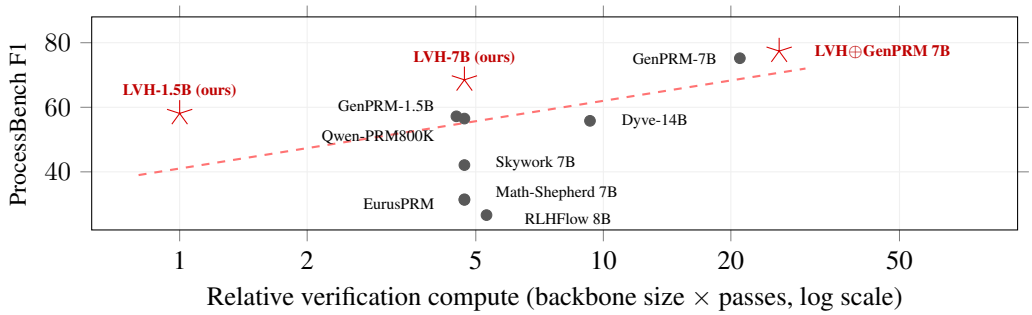


Figure 1: Introspective verification is compute efficient. Mean ProcessBench localization F1 against the relative compute of one verification, taken as the backbone size times the number of forward passes on a log scale, a rough proxy (the token and wall clock accounting is in Section 4.3). The introspective head (stars) sits above the trend of trained process reward models at both scales. From a single forward pass and no generation, LVH-1.5B beats the classification models outright and matches a PRM800K model at 7B and Dyve at 14B, and LVH-7B reaches 68.5 above every same cost baseline. Fused with a generative verifier it passes GenPRM-7B at a small added cost.

A separate line of evidence points to a cheaper source of the same judgment. A model's hidden states already encode whether its output is correct, which linear probes recover for factual statements and for answer accuracy (Azaria & Mitchell, 2023; Kadavath et al., 2022; Burns et al., 2023; Marks & Tegmark, 2023; Orgad et al., 2025; Liu et al., 2024). Recent work turns these internal signals into rewards for Best of N selection (Guo et al., 2026), aligns hidden states to improve reinforcement learning (Wang et al., 2026), steers a verifier through its latent representations (Zhou et al., 2026; Hsu & Yao, 2026), and reads step correctness from a frozen backbone (Ni et al., 2026). This evidence has been used mostly for answer level truthfulness, for selection, or for steering. Whether a discriminative read of the hidden states can match a generative process reward model on step level verification, at a fraction of its cost, has not been shown.

We show that this read is possible and competitive. A linear probe on a single intermediate layer of a frozen backbone separates correct from incorrect steps well above chance, and the signal is strongest near seventy percent of the depth rather than at the final layer. We call this operation introspection. The verdict for a step is decoded from the backbone's own intermediate representation, instead of being generated as a critique. This probed signal is stronger at the intermediate layer than at the final layer, and a structural explanation is consistent with the pattern, though we do not test it with causal interventions. The final state is optimized to emit the surface verdict token, so it could be misled by a fluent but wrong step, while an earlier state is formed before that token is produced and may align more closely with whether the step follows from its premises. This is a statement about how cleanly a probe can separate correct from incorrect steps in each layer's untrained representations. Once a readout is trained on those representations, the final layer becomes the stronger single view, and our head combines the two (Appendix A.3).

We build an introspective verdict head that sums two small readouts, one on an intermediate layer and one on the final layer, and returns a per step reward in a single forward pass, trained with LoRA on a small slice (about eight thousand conversations) of one released label set and fitting on a single 32GB V100 within a few hours at both the 1.5B and the 7B scale (Figure 1). Once trained, the intermediate readout alone is the weaker single view, and the final readout alone is strong on easier problems but weaker on proof heavy ones. Their sum matches the stronger readout in mean F1 and improves the proof heavy subsets above either alone. On ProcessBench, the 1.5B head reaches 58.1 mean F1 from one forward pass, exceeding most trained process reward models, including ones at 7B to 14B scale, and a same size generative verifier, while approaching GPT-4o's 61.9 at about a quarter of one generative pass' token cost. The two signals are complementary: a logit space combination with a generative verifier, adds 5.4 mean F1 at 1.5B, and at 7B, where the generative verifier alone is stronger, an unsupervised per domain calibration raises mean F1 from 75.2 to 77.4 without lowering any subset.

Our contributions are the following.

- We identify introspection as a mechanism for process verification. Step correctness is

decodable by a probe on the intermediate hidden states of a verifier backbone, where the intermediate layer carries a stronger signal than the final layer.

- We build an introspective verdict head, abbreviated LVH, a residual classifier over an intermediate and a final layer that produces per step rewards in one forward pass, trained with LoRA on step labels within a few hours on a single 32GB V100, at both the 1.5B and the 7B scale.
- On ProcessBench at 1.5B, introspection exceeds most trained process reward models on the benchmark, including models at 7B and 14B scale, from a single forward pass that costs about one quarter of one generative pass, and it approaches GPT-4o, which uses no process reward model.
- A logit space combination of introspection with a generative verifier, always fit on cases disjoint from the ones it scores, improves the generative verifier itself. It adds 5.4 mean F1 at 1.5B, and at 7B, and with an unsupervised per domain calibration, it raises mean F1 from 75.2 to 77.4 without lowering any subset.

2 Related Work

Outcome verifiers score a full solution (Cobbe et al., 2021). Process reward models score each step instead (Uesato et al., 2022; Lightman et al., 2024), with labels from human annotation (Lightman et al., 2024), Monte Carlo rollouts (Wang et al., 2024), or consensus filtering (Zhang et al., 2025b). ProcessBench standardizes this evaluation and finds classification models score below strong critics on proof-heavy subsets (Zheng et al., 2025a), per a recent survey (Zheng et al., 2025b) and later safety-critical extensions (Pronesti et al., 2026). Other work lowers this cost: log-ratio implicit rewards (Yuan et al., 2025), answer-derived pseudo-labels (Sun et al., 2025), reverse-verification labels (Tan et al., 2026), and a shared-backbone scoring head (Wang et al., 2025). All read a scalar from the final layer or from generation. Our head reads an automatically selected intermediate layer and combines it with the final layer through a residual.

The most accurate verifiers score by generation: a critique as next-token prediction (Zhang et al., 2025a), reasoning before judging (Xiong et al., 2025), or generating and running code (Zhao et al., 2026), at a cost that grows with critique length and sample count (Snell et al., 2024; Zhao et al., 2026). Dyve routes easy steps to a fast check to lower the average (Zhong et al., 2025). Our introspection based method removes generation entirely, keeping one forward pass, and is complementary: a logit space combination, always fit on cases disjoint from the ones it scores, improves a strong generative model on every ProcessBench subset.

Hidden states encode whether a model's own output is correct: linear probes recover this for factual statements (Azaria & Mitchell, 2023; Kadavath et al., 2022; Burns et al., 2023; Marks & Tegmark, 2023), steering along these directions shifts truthfulness (Li et al., 2023), and correctness is predictable internally, sometimes before the answer appears (Orgad et al., 2025; Liu et al., 2024; Cencerrado et al., 2025). Closest to us, such signals become Best-of-N rewards (Guo et al., 2026) or a frozen-backbone step probe (Ni et al., 2026), targeting answer-level truthfulness or selection rather than step-level verification. Linear classifiers routinely locate information across depth (Alain & Bengio, 2017), and lenses map intermediate states to output distributions (Belrose et al., 2023), motivating our reading depth. We adapt the backbone with LoRA (Hu et al., 2022), select that depth from the model's own measurement, combine intermediate and final layers, and at the same backbone size match generative accuracy at a fraction of the compute. Test-time self-consistency uses such verifiers to select among sampled solutions (Wang et al., 2023).

3 Method

We describe the introspective verdict head, which reads a per step reward from the backbone's hidden states in a single forward pass. We fix notation (Section 3.1), define the introspective head (Section 3.2), explain the layer choice (Section 3.3), state the training objective (Section 3.5), and give the inference procedure and its cost (Section 3.6). An optional logit space fusion with a generative verifier is in Section 3.7.

3.1 SETUP AND NOTATION

A problem x has a candidate solution written as an ordered sequence of steps $s_1, \ldots, s_k$. A process verifier assigns each step a reward $r_i \in [0, 1]$, where a reward above a threshold marks the step as an error, and the first such step is the located error. ProcessBench scores this located first error against the labeled one.

The backbone is a decoder only transformer f with L layers. For an input token sequence it returns hidden states $h^{(0)}, \ldots, h^{(L)}$, where $h^{(0)}$ is the embedding and $h^{(L)}$ is the final layer. We write $h_i^{(\ell)} \in \mathbb{R}^d$ for the hidden state at layer ℓ at the last token of step s_i, that is the token that ends the step in the concatenation $x, s_1, \ldots, s_k$. Because attention is causal, this last token has attended to all of x and $s_1, \ldots, s_i$, so its hidden state is a natural summary of the step in the context that precedes it. Reading a per step reward from $h_i^{(\ell)}$ therefore requires no extra generation, only the states that a single pass over the solution already produces.

3.2 THE INTROSPECTIVE VERDICT HEAD

The introspective head sums two readouts, one on the final layer and one on an intermediate layer $\ell^\star$,

$$r_i = \sigma\Big(\underbrace{\mathbf{w}^\top h_i^{(L)}}_{\text{stated verdict}} + \underbrace{g\big(h_i^{(\ell^\star)}\big)}_{\text{latent assessment}} \Big), \qquad z_i \triangleq \mathbf{w}^\top h_i^{(L)} + g\big(h_i^{(\ell^\star)}\big), \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is a linear map on the final layer, $g : \mathbb{R}^d \to \mathbb{R}$ is a two layer perceptron with a GELU nonlinearity ($\mathbb{R}^d \to \mathbb{R}^m \to \mathbb{R}$, with $m = 512$), σ is the logistic function, and z_i is the pre sigmoid logit of the reward. The two readouts play different roles, which we name descriptively rather than as a claim of a clean causal decomposition. The final layer readout follows the verbalized verdict that the backbone would emit, so we call it the stated view, presentation oriented and prone to fluent but wrong steps. The intermediate readout is taken before that surface token is formed, so we call it the latent view, the assessment behind the answer. Their residual sum keeps the calibration of the stated view and the discrimination of the latent view, and Figure 2 contrasts this single pass read with a generative verifier that decodes a critique, runs code, and votes over samples.

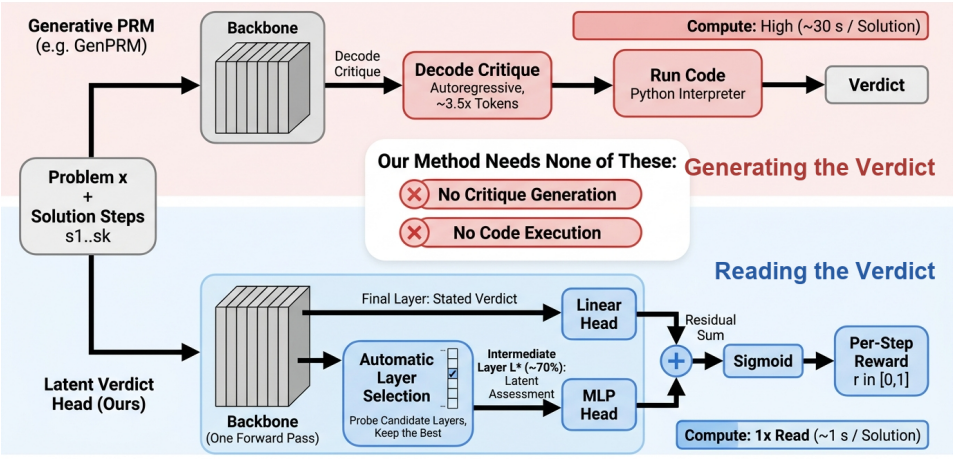


Figure 2: Reading the verdict instead of generating it. Top: a generative process reward model decodes a natural language critique, runs code, and often votes over several samples before a verdict, which costs many decoded tokens and an interpreter call. Bottom: the introspective verdict head reads the same solution in a single forward pass, takes the stated verdict from the final layer and the latent assessment from an intermediate layer $\ell^\star$, and sums them into a per step reward. It needs no critique generation and no code execution, at about one quarter of the token cost of a single generative pass. Both methods use the same backbone and differ only in what follows it, and the intermediate layer $\ell^\star$ is selected automatically by a probe over candidate layers (Algorithm 2 in Appendix).

3.3 WHY AN INTERMEDIATE LAYER

We place $\ell^\star$ at about seventy percent of the depth, eight layers below the final layer (layer 20 of 28), which a per layer probe sweep selects (Figure 3a). A structural explanation is consistent with this choice, though we test it only with probe separability and the ablation of Appendix A.3, not causal interventions. The final state is optimized to emit the surface verdict token, so it could be moved by a fluently wrong step, while a state a few layers earlier, formed before that token, may align more closely with whether the step follows from its premises. Section 4.3 shows step correctness is in fact most separable at this depth and weaker at the final layer.

3.4 AUTOMATIC LAYER SELECTION

We do not fix $\ell^\star$ by hand. Before training, a cheap probe selects it from a small labeled set, as in Algorithm 2 (Appendix A.1). For each candidate layer we fit a class balanced logistic probe on the last token hidden state of every step and keep the candidate whose probe localizes the first error best on a held out split. The candidate set $\mathcal{C}$ is the final layer and every second layer below it, eight in all, and the labeled set is a few hundred solutions from the training distribution. The procedure costs one forward pass per solution, since a single pass returns every candidate layer at once, and it passes no gradient through the backbone and needs no labels beyond those training already uses. We run the selection on the 7B backbone (Section 4.3), where it returns the layer eight below the final layer, near seventy percent of the depth, and since both backbones have 28 layers we apply the same relative depth $\ell^\star = 20$ of 28 at 1.5B. The training below then adapts the introspective head at the selected $\ell^\star$.

3.5 TRAINING

We adapt the backbone with LoRA (Hu et al., 2022) of rank 16 and scaling 32 on the query, key, value, and output projections, and train the LoRA parameters and the two heads jointly. The rest of the backbone is frozen, and gradient checkpointing keeps the backbone and the heads within a single 32GB V100 at both the 1.5B and the 7B scale. Supervision is a set of step level labels $\mathcal{D} = \{(x, s_{1:k}, y_{1:k})\}$ with $y_i \in \{0, 1\}$, where $y_i = 1$ marks an incorrect step. We minimize a class balanced binary cross entropy on the step logits of Equation (1),

$$\mathcal{L} \;=\; -\sum_{(x,s_{1:k},y_{1:k})\in\mathcal{D}} \;\sum_{i=1}^{k} \Big[\, w\, y_i \log \sigma(z_i) \;+\; (1-y_i) \log\big(1-\sigma(z_i)\big)\Big], \qquad w \;=\; \frac{1-p}{p}, \quad (2)$$

where p is the fraction of error steps in $\mathcal{D}$. The weight w multiplies only the error term, so when errors are rare and p is small, w is large and the two classes contribute comparably, which offsets the imbalance. The labels come from the step level annotations released with GenPRM (Zhao et al., 2026), which are built from the MATH training set alone. The method uses only these labels, not the generative model that produced them, and only a small slice of them, on the order of eight thousand of the roughly 23 thousand released conversations, none drawn from ProcessBench. Training therefore sees a single mathematical domain, while Section 4 evaluates on four. Only the MATH subset shares a distribution with the training labels. GSM8K, OlympiadBench, and OmniMATH are domains the introspective head never trains on, so three of the four reported subsets measure cross-domain generalization rather than in-domain fit, and no evaluation subset is merely held out at the level of individual solutions.

3.6 INFERENCE AND COST

Inference is one forward pass, given in full in Algorithm 1. Given the problem and its steps we run the backbone once with hidden states, gather the last token index of each step, and apply Equation (1) to obtain $r_1, \ldots, r_k$. The located first error is the first step whose reward exceeds a threshold τ, and τ is fixed on a held out split. There is no autoregressive decoding and no interpreter call. The computation cost is one forward pass, which processes all n tokens of the solution at once. In comparison, a generative verifier reads the same solution and then decodes a critique several times longer than the solution itself, about 3.5 times as many tokens on our 7B outputs, which costs one further forward pass per generated token, and it may also run code. Section 4.3 quantifies the gap.

Algorithm 1 Introspective scoring of a k step solution (inference)

Require: problem x, steps $s_1, \ldots, s_k$, adapted backbone f, heads $\mathbf{w}, g$, layer $\ell^\star$, threshold τ
1: Tokenize $x, s_1, \ldots, s_k$ and record the last token index t_i of each step
2: $H \leftarrow f(x, s_{1:k})$ // one forward pass, hidden states at all layers, no decoding
3: **for** $i = 1$ to k **do**
4: $\quad z_i \leftarrow \mathbf{w}^\top H_{t_i}^{(L)} + g\big(H_{t_i}^{(\ell^\star)}\big), \quad r_i \leftarrow \sigma(z_i)$
5: **end for**
6: $e \leftarrow \min\{\, i : r_i > \tau \,\}$, or none if no step exceeds τ
7: **return** rewards $r_{1:k}$ and located first error e

3.7 Fusion with a generative verifier

The introspective reward and a generative verifier read different evidence, so they can be combined. Given a generative verifier's per step correctness probability P_i^B, we form a logit space score

$$\zeta_i \;=\; a\,\big(z_i - c\big) \;+\; (1-a)\,\mathrm{logit}\big(1 - P_i^B\big), \tag{3}$$

and locate the first error at the first step with ζ_i above a bias b. Here z_i is the introspective head's step logit from Equation (1) and $\mathrm{logit}(p) = \log \frac{p}{1-p}$. Because P_i^B is a correctness probability, both terms are log odds that step i is an error, so they share a common scale. The mixing coefficient $a \in [0, 1]$ sets how much of the decision comes from introspection. Both a and b are always fixed on held out data and never on the subset being scored. They can be fixed once on a single held out split and applied unchanged to every subset, which is the simpler setting, or refit for each subset on the remaining subsets, a leave one domain out protocol that gives stronger performance. Section 4.1 states which protocol each reported result uses. The scalar c is an unsupervised per domain centering that we apply only at the 7B scale, and we explain in Appendix A.4 why the 7B head needs it and the 1.5B head does not. The fusion adds one forward pass to the generative pipeline, which is negligible compared with the one pass per generated token that decoding already requires.

4 Experiments

This section evaluates two claims about the introspective head. The first is that at 1.5B a discriminative read of the backbone's hidden states is competitive with a generative process reward model of the same size, at a fraction of its cost. The second is that at both scales this read carries evidence the generative verifier does not already have, so combining the two improves the generative verifier. Table 1 reports the main results. The rest of this section describes the setup, then reports the layer mechanism, the choice of decision threshold, the fusion, and the performance at 7B. At 7B the standalone head falls short of the larger generative verifier, so fusion rather than replacement is the claim the evidence supports there.

4.1 Setup

Benchmark and metric. We evaluate on ProcessBench (Zheng et al., 2025a), which asks a verifier to locate the first incorrect step in a solution, or to report that every step is correct. Its four subsets, in increasing difficulty, are GSM8K, MATH, OlympiadBench, and OmniMATH. We call one problem with its candidate solution a case, so a case spans many steps but receives a single verdict, and every count of cases below is a count of solutions rather than steps. The metric per subset is the localization F1, the harmonic mean of accuracy on erroneous solutions, where the predicted first error must match the labeled step, and accuracy on correct solutions, where the verifier must predict no error.

Backbones. We use DeepSeek-R1-Distill-Qwen at the 1.5B and the 7B scale as the verifier backbone. Both have 28 transformer layers, so $\ell^\star = 20$ is the same intermediate layer at both scales. Training and inference run on a single 32GB V100 in a few hours, in bf16 or fp16, with gradient checkpointing so the backbone and the two heads stay within one device.

Training labels. Supervision is the step level labels released with GenPRM (Zhao et al., 2026), drawn from MATH training problems alone as described in Section 3.5. No ProcessBench case contributes a training label, so the four subsets serve purely as held out validation, and the three other

Table 1: ProcessBench localization F1. *Top panel:* 1.5B scale. *Bottom panel:* 7B scale. All baseline rows, including the GenPRM greedy row, are released scores from ProcessBench (Zheng et al., 2025a) and GenPRM (Zhao et al., 2026) Table 1, and only our own rows, the introspective verdict head (LVH) of Section 3 and its fusion, are measured in our environment. Introspection uses one forward pass, with no generation and no code execution. The 1.5B fusion row is measured on 120 cases per subset, where the same generative verifier scores 58.3, so its gain is the 5.4 points of Section 4.5 rather than a comparison with the released row above. LVH-7B standalone uses the unsupervised per domain calibration of Section 4.6, and the fusion row is the honest leave one domain out combination.

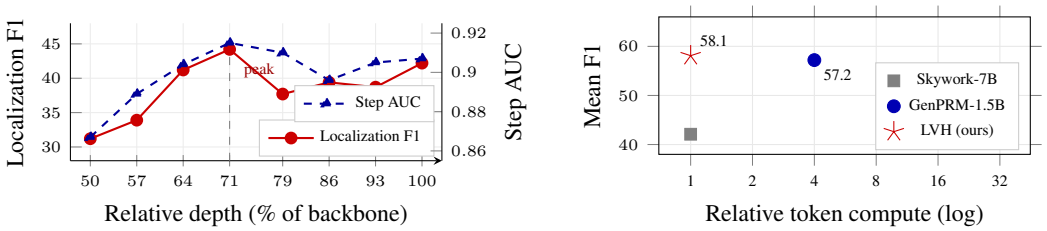
	Method	GSM8K	MATH	Olympiad	OmniMATH	Mean
1.5B	Math-Shepherd-PRM-7B	47.9	29.5	24.8	23.8	31.5
	RLHFlow-PRM-8B	38.8	33.8	16.9	16.9	26.6
	Skywork-PRM-7B	70.8	53.6	22.9	21.0	42.1
	EurusPRM-Stage2	47.3	35.7	21.2	20.9	31.3
	Qwen2.5-Math-7B-PRM800K	68.2	62.6	50.7	44.3	56.5
	Dyve-14B	68.5	58.3	49.0	47.2	55.8
	GPT-4o (critic, no PRM)	79.2	63.6	51.4	53.5	61.9
	GenPRM-1.5B (greedy)	52.8	66.6	55.1	54.5	57.2
	LVH-1.5B (ours, one pass)	**65.1**	61.3	53.5	52.4	**58.1**
	LVH $\oplus$ GenPRM-1.5B (fusion)	**64.5**	**73.0**	**61.7**	**55.4**	**63.7**
7B	GenPRM-7B (greedy)	78.7	80.3	72.2	69.8	75.2
	LVH-7B (ours, one pass)	72.5	76.6	68.1	57.0	68.5
	LVH $\oplus$ GenPRM-7B (fusion)	**83.4**	**81.5**	**73.2**	**71.5**	**77.4**

than MATH are also out of the training domain. Two evaluation-time adjustments touch ProcessBench data without touching any parameter, plus one that does not touch ProcessBench at all. The decision threshold (Section 4.4) is fixed on a held-out slice of the same released label set used for training, not on ProcessBench. The fusion hyperparameters, the mixing coefficient a and bias b of Equation (3), are set differently at the two scales. At 1.5B a single pair is fixed on a disjoint held-out ProcessBench slice and applied unchanged to all four subsets. At 7B a separate pair is fit for each subset from the other three only, so the pair that scores GSM8K is fit on MATH, OlympiadBench, and OmniMATH.

Baselines. We compare against the process reward models tabulated on ProcessBench and by GenPRM, named individually in Table 1. They span four classification models (Wang et al., 2024), a discriminative model trained on PRM800K (Lightman et al., 2024; Zhang et al., 2025b), adaptive routing at 14B (Zhong et al., 2025), GPT-4o used as a critic without a process reward model, and GenPRM at the matching 1.5B and 7B size in its greedy single pass form.

4.2 Introspection is competitive with a generative verifier

Table 1 (top panel) reports the ProcessBench result at the 1.5B scale. The introspective head reaches 58.1 mean F1 from a single forward pass, with a case level bootstrap 95% interval of $[56.0, 59.9]$. This exceeds every classification process reward model in the table, the discriminative PRM800K model at 7B, and Dyve at 14B. At the mean level it also matches or slightly exceeds the same size generative model GenPRM-1.5B in its greedy form, 58.1 against 57.2, while reading the solution once. That is one forward pass and no interpreter call, against the same read plus one pass per decoded token and an interpreter call for the generative verifier. Introspection therefore spends about one quarter as much token compute, and the wall clock gap is even larger (Appendix B.2). The score approaches GPT-4o's 61.9. Introspection leads by a wide margin on GSM8K, 65.1 against 52.8, where first errors are concrete slips that a latent read catches directly. It scores below GenPRM-1.5B on the other three subsets, by 5.3 points on MATH, 1.6 on OlympiadBench, and 2.1 on OmniMATH, where GenPRM-1.5B often needs re-deriving the step. These are narrow margins for what introspection does not do. On both proof heavy subsets it stays within about two points of a generative verifier while decoding no critique, calling no interpreter, and spending a quarter of the compute, so the split marks where a cheap read suffices and where generation earns its cost.



(a) Per layer probe separability (7B).

(b) Cost–accuracy frontier (1.5B).

Figure 3: Where the signal lives and what it costs. **(a)** Step level AUC (right axis) and probe localization F1 (left axis) both peak at about seventy percent of the depth. **(b)** The introspective head (star) reaches higher mean F1 than a greedy generative pass at about one quarter of its token compute (GPT-4o, 61.9, omitted as a much larger model).

4.3 An intermediate layer carries the signal

Before the benchmark we locate where step correctness is decodable. We freeze the 7B backbone, take the last token hidden state of each step at eight candidate layers, and fit a balanced logistic probe on step correctness. Figure 3(a) plots the step level AUC and the probe's localization F1 per layer. Full numbers are in Table 2 (Appendix A.2). Both peak at the same place: the intermediate layer eight below the final layer, about seventy percent of the depth, at 0.915 AUC and 44.2 F1, declining toward the final layer and toward the middle of the network. So in raw separability the intermediate layer, not the final layer, reads out step correctness most cleanly, and this is visible from a cheap linear probe before the LoRA-adapted head is trained at all, a property of the backbone itself rather than something training creates. This sweep is exactly the automatic selection of Algorithm 2 run on the 7B backbone, so introspection reads from the layer this procedure returns. The read is also cheap: Figure 3(b) shows the 1.5B head matches or beats compute-matched baselines at a quarter of one generative pass' token cost, and Table 1 shows this same score exceeds several larger trained verifiers, including ones at 7B and 14B scale. At 7B this is a median wall clock speedup near 32 times over the greedy generative verifier (Appendix B.2 gives the full accounting).

A final layer readout is added to the intermediate one because the intermediate readout ranks steps well but produces scores whose absolute values are arbitrary, so no single threshold suits it alone. Adding the final layer readout keeps that ranking and puts the sum on a scale where one fixed threshold works. Appendix A.3 shows this combination is what lets a head trained only on MATH labels generalize its discrimination to the two subsets furthest from that training distribution, OlympiadBench and OmniMATH, where either readout alone falls short.

4.4 A single decision threshold transfers across subsets

Introspection returns a per step probability, and a single threshold turns it into a decision. Figure 4 sweeps this threshold across all four subsets at 1.5B. GSM8K is the only subset that peaks and then declines, reaching its best near 0.87 and falling away above it. The other three rise with τ and reach their best at the top of the swept range, since a higher bar removes false alarms where introspection over flags. They differ in how quickly they saturate. MATH is within half a point of its best from $\tau = 0.75$ onward, while OlympiadBench and OmniMATH are still six and five points below theirs at that threshold. The two ends of the benchmark therefore prefer different thresholds, so the single value used throughout, 0.92, is a compromise rather than a threshold insensitive choice. It nonetheless lands within about two points of the best plotted value on every subset, so one fixed threshold transfers to all four without per subset tuning. This per subset disagreement is also the motivation for the unsupervised per domain calibration of Appendix A.4 at 7B.

4.5 Introspection is complementary to generation

The two signals read different evidence, so we test whether combining them locates more first errors than either one alone. We form a logit space sum of the introspective reward and a generative verifier score, with a single mixing coefficient and bias fixed on a held out MATH split and never on the test subsets. On 120 matched cases from each subset, this fixed combination improves the mean

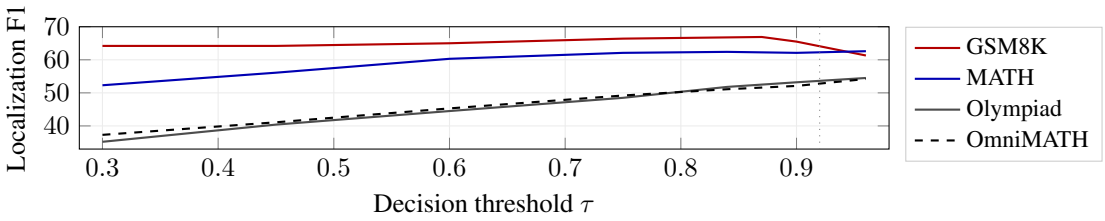
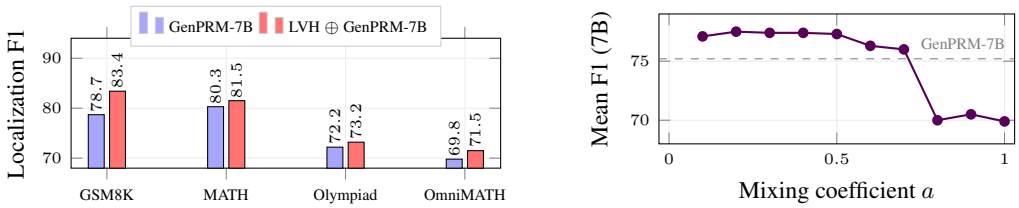


Figure 4: Localization F1 against the decision threshold for LVH-1.5B, on all four ProcessBench subsets. GSM8K alone peaks and then declines, near $\tau = 0.87$. The other three rise with τ and are best at the top of the swept range, MATH saturating early while Olympiad and OmniMATH are still several points below their best at $\tau = 0.75$. The dotted line is the single held out tuned threshold (0.92), used for the 1.5B LVH rows of Table 1. The 7B rows use the separate per domain calibration of Section 4.6.



(a) Fusion decomposition by subset.

(b) Robust to the mixing coefficient.

Figure 5: Complementarity at 7B. **(a)** The honest leave one domain out fusion matches or beats the generative verifier on every subset. **(b)** Fusion mean F1 against the mixing coefficient a. A broad plateau spans $a \in [0.1, 0.5]$, above the generative-only baseline (dashed), so the gain is insensitive to the exact coefficient.

localization F1 by 5.4 points over the greedy generative verifier alone. The gain is not a re-ranking of the same errors. Introspection recovers cases where a fluent but wrong step misleads the surface verdict, and the generative verifier keeps the cases where a written derivation is needed, so the sum is above either input. An introspective read is therefore not only cheap, it carries evidence a generative verifier does not already have.

4.6 Fusion achieves better performance at 7B model

We repeat the study at 7B on 120 cases per subset, in the bottom panel of Table 1. The honest leave one domain out fusion, never tuned on the domain it scores, raises the mean from 75.2 to 77.4 without lowering any subset, using a per domain calibration that needs no labels (Appendix A.4). The one extra forward pass that fusion adds is negligible compared with the many decoding passes the generative verifier already spends. A case level bootstrap (3000 resamples) gives 95% intervals for our own two rows of Table 1, LVH-7B standalone and the fusion. The standalone 68.5 has an interval of $[63.3, 73.0]$ and the fusion's 77.4 has $[73.4, 81.1]$. Figure 5a decomposes the 7B fusion by subset, where it beats the generative verifier on all the datasets and gains most on GSM8K and OmniMATH, and Figure 5b shows the gain is insensitive to the exact mixing coefficient.

5 Conclusion

We asked whether a generative process reward model's judgment is already available in the verifier's hidden states, without generation, and found that much of it is. Step correctness is decodable by a probe on an intermediate layer near seventy percent of the depth, more strongly than from the final layer (Section 3.3). We built an introspective verdict head, a residual readout over that layer and the final one, trained with LoRA on roughly eight thousand conversations and evaluated in one forward pass on a single 32GB V100. At 1.5B it reaches 58.1 mean F1, above most trained process reward models including ones at 7B and 14B scale and at an order of magnitude less compute, and fusing it with a generative verifier improves that verifier at both scales. Appendix D shows the limitations of our method on calibration, coarse localization, and errors of omission.

Reproducibility Statement

The frozen backbone (DeepSeek-R1-Distill-Qwen, 1.5B and 7B), the LoRA configuration (rank 16, scaling 32 on the query, key, value, and output projections), the head architecture (Equation (1)), and the training objective (Equation (2)) are given in full in Section 3. The training label source, its size, and the evaluation protocol are described in Section 4.1. The inference procedure is given as pseudocode in Algorithm 1, and the automatic layer selection procedure in Algorithm 2 (Appendix A.1). All baseline numbers are released scores cited to their source (Table 1), and every number we measured ourselves is annotated as such throughout. Training and inference both run on a single 32GB V100, so the full pipeline is reproducible on modest hardware. Code is available at `https://anonymous.4open.science/r/LLM_CALM-48B7`.

AI Use Statement

In this work, we used generative AI tools to assist with drafting and editing paper text, formatting citations. We have not used generative AI tools to design the method, run experiments, or generate results; every number we report as our own measurement was produced by our own experiments, not by an AI tool, and baseline numbers are released scores cited to their source rather than measurements of our own. We have reviewed all AI-assisted text for accuracy.

References

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR), Workshop Track*, 2017.

Amos Azaria and Tom M. Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

Iván Vicente Moreno Cencerrado, Arnau Padrés Masdemont, Anton Gonzalvez Hawthorne, David Demitri Africa, and Lorenzo Pacchiardi. No answer needed: Predicting LLM answer accuracy from question-only linear probes. *arXiv preprint arXiv:2509.10625*, 2025.

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

Jizhou Guo, Zhaomin Wu, Hanchen Yang, and Philip S. Yu. Mining intrinsic rewards from LLM hidden states for efficient best-of-n sampling. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 371–382, 2026.

Chia-Hsuan Hsu and Jui-Ming Yao. Learning to refine hidden states for reliable LLM reasoning. *arXiv preprint arXiv:2606.17524*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt,

Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *Transactions on Machine Learning Research*, 2026.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

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 *International Conference on Learning Representations (ICLR)*, 2024.

Junteng Liu, Shiqi Chen, Yu Cheng, and Junxian He. On the universal truthfulness hyperplane inside llms. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 18199–18224, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*, 2023.

Jingwei Ni, Ekaterina Fadeeva, Tianyi Wu, Mubashara Akhtar, Jiaheng Zhang, Elliott Ash, Markus Leippold, Timothy Baldwin, See-Kiong Ng, Artem Shelmanov, and Mrinmaya Sachan. ReProbe: Efficient test-time scaling of multi-step reasoning by probing internal states of large language models. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 11667–11689, 2026.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*, 2025.

Massimiliano Pronesti, Anya Belz, and Yufang Hou. Beyond outcome verification: Verifiable process reward models for structured reasoning. In *Findings of the Association for Computational Linguistics*, pp. 32187–32202, 2026.

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.

Lin Sun, Chuang Liu, Xiaofeng Ma, Tao Yang, Weijia Lu, and Ning Wu. Freeprm: Training process reward models without ground truth process labels. *arXiv preprint arXiv:2506.03570*, 2025.

Xiaoyu Tan, Tianchu Yao, Chao Qu, Bin Li, Minghao Yang, Dakuan Lu, Haozhe Wang, Yinghui Xu, and Xihe Qiu. AURORA: automated training framework of universal process reward models via ensemble prompting and reverse verification. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1378–1389, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 9426–9439, 2024.

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 *International Conference on Learning Representations (ICLR)*, 2023.

Zixiao Wang, Yuxin Wang, Xiaorui Wang, Mengting Xing, Jie Gao, Jianjun Xu, Guangcan Liu, Chenhui Jin, Zhuo Wang, Shengzhuo Zhang, and Hongtao Xie. Test-time scaling with reflective generative model. *arXiv preprint arXiv:2507.01951*, 2025.

Ziyue Wang, Aomufei Yuan, Yongfu Zhu, Shuai Dong, Wenpu Liu, Yiran Yao, Weichu Xie, Yuqi Xu, Caoyuan Ma, Wenqi Shao, Xiaoying Zhang, Nan Duan, and Jiaqi Wang. Right makes might: Aligning verified hidden states empowers RL reasoning. *arXiv preprint arXiv:2606.03234*, 2026.

Wei Xiong, Wenting Zhao, Weizhe Yuan, Olga Golovneva, Tong Zhang, Jason Weston, and Sainbayar Sukhbaatar. Stepwiser: Stepwise generative judges for wiser reasoning. *arXiv preprint arXiv:2508.19229*, 2025.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *International Conference on Machine Learning (ICML)*, volume 267, 2025.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 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. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025b.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. Genprm: Scaling test-time compute of process reward models via generative reasoning. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 34932–34940, 2026.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1009–1024, 2025a.

Congming Zheng, Jiachen Zhu, Zhuoying Ou, Yuxiang Chen, Kangning Zhang, Rong Shan, Zeyu Zheng, Mengyue Yang, Jianghao Lin, Yong Yu, and Weinan Zhang. A survey of process reward models: From outcome signals to process supervisions for large language models. *arXiv preprint arXiv:2510.08049*, 2025b.

Jianyuan Zhong, Zeju Li, Zhijian Xu, Xiangyu Wen, and Qiang Xu. Dyve: Thinking fast and slow for dynamic process verification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 22320–22333, 2025.

Yefan Zhou, Yilun Zhou, Austin Xu, Soroush Vosoughi, Shafiq Joty, and Jiang Gui. The hidden signal of verifier strictness: Controlling and improving step-wise verification via selective latent steering. *arXiv preprint arXiv:2605.20745*, 2026.

A Supporting detail

This appendix collects supporting detail and two case studies. Appendix A.1 gives the step by step layer selection procedure, Appendix A.2 gives the full per layer probe numbers behind Figure 3a, Appendix A.3 details the residual-form ablation, Appendix B.1 breaks down where introspection succeeds and fails, Appendix A.4 explains the scale specific median shift used only at 7B, and Appendix A.5 documents the OmniMATH ranking limit. Appendix C works through four real ProcessBench solutions, three where one verifier succeeds and the other fails, and one where both miss the error, showing why the fusion of Section 4.6 helps and where it cannot. Unless stated otherwise the metric is ProcessBench localization F1.

A.1 Layer selection procedure

Algorithm 2 gives the step by step procedure behind Section 3.4. It runs once before training, passes no gradient through the backbone, and returns the intermediate layer $\ell^\star$ whose probe localizes the first error best on a held out split.

Algorithm 2 Automatic selection of the intermediate layer $\ell^\star$

Require: candidate layers $\mathcal{C}$, labeled set $\mathcal{D}_{\text{sel}}$ with per step labels, split into fit and held out parts, frozen backbone f

1: **for** each candidate layer $\ell \in \mathcal{C}$ **do**
2: $X_\ell \leftarrow$ last token hidden states at layer ℓ for all steps in the fit part
3: fit a class balanced logistic probe p_ℓ on (X_ℓ, y)
4: $F_\ell \leftarrow$ localization F1 of p_ℓ on the held out part
5: **end for**
6: **return** $\ell^\star = \arg\max_{\ell \in \mathcal{C}} F_\ell$

A.2 Per layer probe sweep

Table 2 gives the full numbers behind Figure 3a. We freeze the 7B backbone, take the last token hidden state of each step at eight candidate layers, fit a balanced logistic probe on step correctness, and report the step level AUC on a held out split of the training pool and the probe localization F1 on 140 held out cases. Both peak at about seventy percent of the depth and fall off toward the final layer and toward the middle of the network.

Table 2: Per layer probe separability on the frozen 7B backbone, the numbers behind Figure 3a. Layer $-k$ denotes the k-th layer counting from the output inclusive, so -1 is the final layer (layer 28 of 28) and -9 is eight transformer layers earlier (layer 20 of 28). Relative depth is the position among the 28 transformer layers. Step correctness is most separable, on both the step level AUC and the probe localization F1, at the intermediate layer eight below the final layer, near seventy percent of the depth, and separability falls off toward the final layer.

Layer (from top)	-1	-3	-5	-7	-9	-11	-13	-15
Relative depth	100%	93%	86%	79%	**71%**	64%	57%	50%
Step AUC	0.907	0.905	0.896	0.910	**0.915**	0.904	0.889	0.867
Localization F1	42.2	38.7	39.4	37.7	**44.2**	41.2	33.9	31.2

A.3 The residual form

The introspective head reads the intermediate layer, where step correctness is most separable (Table 2), and adds a final layer readout that we use as a proxy for calibration, inferred from localization F1 rather than a directly measured calibration metric such as ECE. Table 3 trains the two readouts separately and together at the same budget to show what each contributes. A trained readout is not the same as a raw probe, and the two roles are different. The final layer readout is trained to emit the verdict itself, so as a standalone readout it reaches 58.6 mean F1 despite the final layer's raw separability being the lower of the two (Table 2), which we read as a sign it is better matched to the decision rule. The intermediate readout alone is discriminative but reaches only 54.4 on its own, since nothing fixes where its scores sit, so no single threshold suits them. The residual head keeps the intermediate layer's discrimination and the final layer's better-matched scale: it dominates the intermediate readout on every subset, and it exceeds the standalone final readout on the two proof heavy subsets, OlympiadBench and OmniMATH, by 1.9 and 1.7 points, trading away 2.8 and 3.0 points on GSM8K and MATH to get there. The value of the residual form is therefore generalization purchased at a small cost on the easy subsets. It is what lets a head trained only on MATH labels carry more of its discrimination to the subsets furthest from that training distribution. This pattern argues against reading the introspective head as simple distillation of the GenPRM labels into a classifier: a classifier fit purely to match those labels has no structural reason to weight the intermediate layer more heavily, and Section 4.3 shows the intermediate layer's advantage over the final layer is already present in the raw, untrained probe, before the labels shape any readout at all.

A.4 A calibration trick for the 7B fusion

The 7B fusion uses one small unsupervised trick, which we report for completeness rather than as part of the method. The 7B head assigns systematically different logit levels to different domains, so a single global decision threshold is misaligned per domain and is too strict on OlympiadBench in particular. Subtracting each domain's own median logit, which needs no labels, realigns the domains

Table 3: Contribution of each readout at 1.5B, trained at the same full budget (8000 conversations, 15000 training steps, i.e. optimizer updates, not solution steps), localization F1. The final readout is trained to be the verdict, so on its own it reaches a higher F1 despite lower raw separability. The intermediate readout is discriminative but reaches a lower F1 on its own. The residual head keeps both and exceeds the standalone final readout on the two proof heavy subsets.

Trained readout	GSM8K	MATH	Olympiad	OmniMATH
Final layer only	**67.9**	**64.3**	51.6	50.7
Intermediate layer only	59.7	58.2	49.9	49.7
Residual (both)	65.1	61.3	**53.5**	**52.4**

to a common threshold. This raises the standalone 7B head from 64.2 to 68.5 mean F1 and is what lets the honest fusion reach 77.4. The 1.5B head does not show this drift. Its per domain logit levels already sit near a common threshold, so the same shift adds nothing and slightly hurts, since on OmniMATH it removes useful spread. The difference is one of scale, the larger and more specialized backbone separates domains more strongly in its logit scale, so we apply the shift only at 7B.

A.5 The OmniMATH ranking limit

OmniMATH is the proof heavy subset and the hardest for the cheap head. Its per step separability is the lowest of the four subsets, a step level AUC of 0.866 against 0.889 on GSM8K (Table 4), but the sharper limit is localization on correct and on long solutions. Introspection has its lowest correct solution accuracy on OmniMATH, 57.7 percent, so it over flags correct proofs, and its highest wrong step rate, 44.3 percent, so when a proof is wrong it often marks the wrong step. Both effects grow with solution length (Figures 6a and 6b). This is why the standalone head is weakest on OmniMATH in Table 1, and why the fusion defers to the generative verifier there. We treat it as an honest limit of a single forward pass read on long free form proofs.

B Extended quantitative analysis

We analyze introspection in more depth from the per step scores its evaluation runs produce, the same scores behind Table 1. Unless noted the scale is 1.5B and the metric is localization F1 or step level AUC as defined in Section 4.1.

B.1 Where introspection succeeds and fails

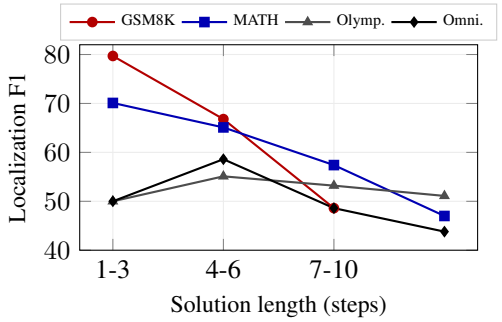
Table 4 splits the localization F1 into its parts, the accuracy on erroneous solutions and on correct ones, and on the error side the fractions where introspection places the error correctly, flags the wrong step, or misses the error. The failure mode shifts with difficulty: on GSM8K introspection mostly misses errors, 25.1 percent, and rarely mislocates them, 21.7 percent, while on OmniMATH it almost never misses, 7.8 percent, but flags the wrong step 44.3 percent of the time. On easy problems introspection fails by overlooking an error, and on hard problems it detects that something is wrong but cannot place it. The step level AUC stays high across subsets, from 0.866 to 0.891, so the difficulty is in localization, not in telling an error step from a correct one in isolation.

Table 4: Per domain error budget for LVH-1.5B, at the tuned threshold. Wrong step and Missed are the fractions of erroneous solutions where introspection flags the wrong step or reports no error. Step AUC pools per step scores against per step labels.

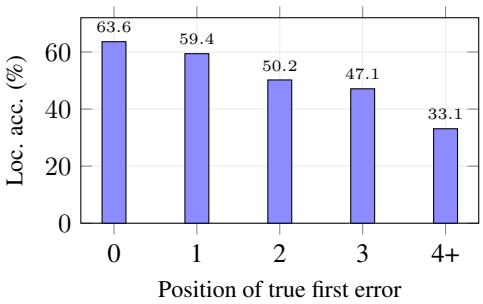
Subset	Step AUC	F1	Error acc.	Correct acc.	Wrong step %	Missed %
GSM8K	0.889	65.1	53.1	83.9	21.7	25.1
MATH	0.891	61.3	51.0	76.8	38.6	10.4
Olympiad	0.876	53.5	46.7	62.5	43.6	9.7
OmniMATH	0.866	52.4	48.0	57.7	44.3	7.8

Two structural factors deepen this localization gap. Localization F1 falls with solution length, for example on MATH from 70.1 on short solutions to 47.0 on the longest, and localization accuracy falls

the later the true error sits, from 63.6 percent at the first step to 33.1 percent at the fifth or later. Both hold because introspection must first read every earlier step as correct. Figure 6 plots both trends in full.



(a) F1 by solution length.



(b) Accuracy by error position.

Figure 6: Localization for LVH-1.5B degrades with solution length **(a)** and with a later error position **(b)**, because introspection must first pass every earlier step.

B.2 Compute accounting

We make the cost gap of Section 4.3 explicit on two axes, token compute and measured wall clock. The introspective head reads the solution once and produces every step reward in that single forward pass, with no autoregressive decoding and no interpreter call. A single greedy pass of the same size generative verifier reads the same solution and then decodes a critique, and it also runs code. On our 7B outputs the critique is about 3.5 times as many tokens as the solution, measured over 480 cases. In token compute, reading is one forward pass over the whole solution while decoding is one forward pass per generated token, so a single greedy generative pass costs about four times the token compute of introspection, and introspection spends about one quarter of it. Wall clock makes the gap larger, because the decoded tokens are produced one forward pass at a time while introspection reads all tokens together in a single pass. On one V100 at 7B, the greedy generative verifier takes a median of 31 seconds per solution end to end including code execution, while the introspective head scores the same solution in about 1 second unbatched, a median speedup near 32 times. We compare against the greedy verifier throughout, since sampling and voting would raise the cost of both verifiers. Introspection therefore sits at a different point on the cost accuracy curve. It reaches a mean F1 in the range of a strong single pass generative verifier while spending the compute of a single read. Figure 3b in Section 4.3 plots the 1.5B cost-accuracy frontier this accounting supports, and Figure 5b in Section 4.6 shows the 7B mixing coefficient is similarly forgiving.

B.3 Calibration

Introspection returns a probability, but Figure 7 shows that this probability does not match the empirical error rate. It is over confident, with an expected calibration error of 0.238, and steps to which it assigns a probability near one are in fact error steps only 41 percent of the time. This is why inference tunes a single threshold rather than reading the probability at 0.5, and it is consistent with the per domain offset that the 7B head shows in Appendix A.4. The ranking is still useful, as the high step level AUC in Table 4 shows, but the raw probability should not be read as a calibrated confidence.

C Case analysis

Table 5 cross-tabulates the two verifiers' verdicts over all the cases they share at 7B. The four boxes below are individual examples drawn from three of its cells, two where introspection alone is right, one where the generative verifier alone is right, and one where both are wrong. Each box below gives the problem and its solution steps, highlights the labeled first error in the text, and records whether the generative verifier (**Gen**) and introspection (**Head**) judge each step correct (✓) or an error (×). Gen flags a step once its correctness value falls below one half, while introspection flags a step once

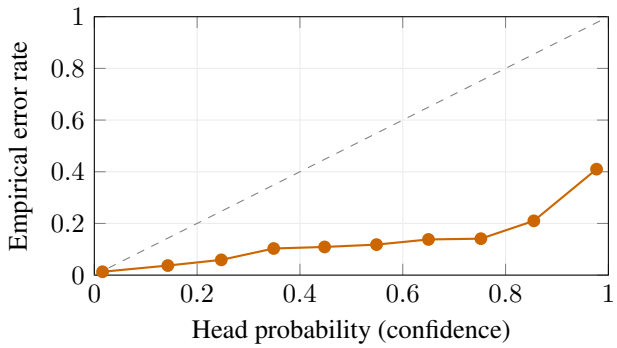


Figure 7: Reliability of LVH-1.5B. Introspection is over confident, the curve sits well below the diagonal, so a high score does not mean a high error probability (ECE $= 0.238$).

its error probability exceeds its 7B operating threshold of about 0.75.

Table 5: Agreement between the two 7B verifiers on their 480 shared evaluation cases, 120 per subset, under a simple unweighted comparison rather than the tuned fusion of Section 4.6. A verdict is right when it matches the label, including when both report no error. The two agree on 342 cases, 275 with both verdicts right and 67 with both wrong, and disagree on the remaining 138. The generative verifier is right on 98 of those and introspection on the other 40. Those 40 are the headroom a fusion can recover, and are not a case level accounting of the tuned fusion's 77.4 mean F1.

	Gen verdict	
Head verdict	right	wrong
right	275	**40**
wrong	**98**	67

Case 1 (GSM8K). Introspection catches a fluent but wrong step that generation accepts.

Problem. It takes James 10 minutes to read 3 pages before bed. He reads 18 pages, then goes to sleep. How long does James spend reading? *(Correct:* $18 \div 0.3 = 60$ *minutes.)*

#	Step	Gen	Head
1	Set up: find a reading rate, then use it	✓	✓
2	Rate $= 3/10 = 0.3$ pages per minute	✓	✓
3	Total pages read $= 18$	✓	✓
4	**Time** $= 18 \times 0.3 = 5.4$ **minutes** *(first error: needs $\div$, not $\times$)*	✓	×
5	Answer: 5.4 minutes	✓	×

Gold: step 4. **Gen predicts:** none (misses the error). **Head predicts: step 4 (correct).**

Why. The two readouts diverge here. The final layer state follows the surface form of the step, and $18 \times 0.3 = 5.4$ is a fluent, locally valid multiplication, so the generated verdict accepts it. The intermediate layer encodes the step against its premises, where the rate is pages per minute and the question asks for minutes, so the correct operation is division. That mismatch is present in the intermediate representation before the surface token is formed, and the residual head reads it out. This is the paper's mechanism in one example, the latent assessment catches a reasoning error that the stated verdict does not.

Case 2 (GSM8K). Introspection avoids a false alarm that generation raises on a correct solution.

Problem. A two hour concert gives each group 2 minutes on, 6 to perform, 2 off, plus one 10 minute intermission. How many groups can perform? *(Correct:* $10n + 10 \leq 120 \Rightarrow 11$ *groups. Every step is correct.)*

#	Step	Gen	Head
1	Fit group slots into 120 minutes	✓	✓
2	Per group $2 + 6 + 2 = 10$ minutes *(correct)*	×	✓
3	Add the 10 minute intermission	×	✓
4	Total for n groups is $10n + 10$	×	✓
5	Solve $10n + 10 \leq 120$	×	✓
6	$n \leq 11$	×	✓
7	Answer: 11 groups	×	✓

Gold: none. **Gen predicts:** step 2 (false alarm). **Head predicts: none (correct)**.

Why. The generative verifier reaches its verdict by re-deriving the step, and its own derivation for the per group time disagrees with the solution, so it reports an error where there is none. Introspection does not re-derive. It reads whether the step is consistent with its premises, and $2 + 6 + 2 = 10$ is, so its error probability stays low on every step. Introspection is less exposed to this self-inflicted false alarm because it does not run a second, fallible generation. This is the complementary win to Case 1, and the reason the two signals combine well (Section 4.6).

Case 3 (MATH). Generation localizes an error that introspection places one step too early.

Problem. Let $x > 0$ satisfy $x - \frac{1}{x} = 3$. Find $x + \frac{1}{x}$. *(Correct:* $x + \frac{1}{x} = \sqrt{13}$*.)*

#	Step	Gen	Head
1	$x^2 = 3x + 1$ *(correct)*	✓	×
2	$x^2 - 3x = 1$**, factor** $(x - 1)(x - 3) = 0$ *(first error: wrong factoring)*	×	×
3	$x = 3$, so $x + 1/x = 10/3$	×	×
4	Answer: 4	×	×

Gold: step 2. **Gen predicts: step 2 (correct)**. **Head predicts:** step 1 (one step early).

Why. Introspection localizes to the region of the error but not the exact step. Its per step probability already saturates at step one, the correct setup that leads straight into the bad factoring, so it fires one step early. The intermediate signal is coarse in position, it marks that the derivation is heading wrong, while the fine boundary between the correct setup and the incorrect factoring is exactly what an explicit step by step re-derivation resolves. The generative verifier performs that re-derivation and pins step two. This coarser localization is the price of a single read, and it is why generation still wins the harder localizations at 7B (Table 5).

Case 4 (GSM8K). Both verifiers miss an omission at the first step.

Problem. In a family of 5, three people eat three eggs each day and the other two eat two each. How many eggs in a week? *(Correct:* $(3 \cdot 3 + 2 \cdot 2) \cdot 7 = 91$*.)*

#	Step	Gen	Head
1	**Count** 3 **people** $\times 3$ **eggs, then** $\times 7$ **days** *(first error: omits the other 2 people)*	✓	✓
2	9 eggs/day $\times 7 = 63$	×	✓
3	63 eggs	×	×

Gold: step 1. **Gen predicts:** step 2 (wrong). **Head predicts:** step 3 (wrong).

Why. The error is one of omission, not of a wrong statement. Step one, count three people times three eggs, is true as far as it goes. What is wrong is the missing contribution of the other two people, which never appears as a locally incorrect step. Both verifiers score each step against its premises, and a step that is locally valid but incomplete passes either test, so both drift to a later step. Errors of omission are a property of the whole solution rather than of one step, so a step local read, generative or introspective, shares this blind spot. These are the 67 both wrong cases of Table 5, and they mark where step level verification itself, not the choice of reading it, reaches its limit.

D LIMITATIONS AND FUTURE WORK

The limits of this method fall into two groups. The first is what a single forward pass can see. Introspection is over confident and needs a tuned threshold, its localization is coarser than an explicit

re-derivation on the hardest subsets, and it cannot catch errors of omission, which are invisible to any step local read. The second is what we did not test. We select $\ell^\star$ once on the 7B backbone and transfer it to 1.5B by matching relative depth (Section 3.4), without sweeping layers at 1.5B, so a scale specific optimum could differ from the transferred choice.

Two directions follow from these limits. A verifier is ultimately measured by the answers it selects, so the first is downstream use, where a reward costing one read is attractive for Best of N selection and for search. The second is to close the localization gap on proof heavy problems, either by reading more than one intermediate layer or by spending a short generation only on the steps where introspection is uncertain, which would keep the average cost near one read while recovering the cases generation currently wins. More broadly, this evidence suggests that a model's assessment of its own reasoning can be read from its states rather than elicited as text, which may hold for judgments beyond step verification.