

# Reasoning Chain Selection via Power Metric Health Signal

## P(t) as a Chain Quality Score on PRM800K: Real Empirical Evidence

*Cole Cantrell | [cole@paradigmbridge.tech](mailto:cole@paradigmbridge.tech) | [paradigmbridge.tech](https://paradigmbridge.tech)*

Paper 18 of an Ongoing Series | [github.com/HauntedKernel/power-metric](https://github.com/HauntedKernel/power-metric)

---

### Abstract

We apply the stochastic power metric  $P(t) = E(t) \times W(t)$  as a chain-level quality signal for reasoning chain selection, evaluated on PRM800K (Lightman et al. 2023) — 30,500 math reasoning chains with human step-level correctness labels.  $P(t)$  computed on human-labeled step-by-step correctness (used here as a proxy signal — real deployment requires a process reward model or confidence proxy) achieves Pearson  $r = 0.955$  with chain quality and 100% in-sample classification accuracy at threshold  $\theta=0.65$ , compared to  $r = 0.529$  and 68.7% accuracy for simple running accuracy. Last-5 step accuracy also achieves 100% in-sample at  $\theta=0.80$ , but relies only on the final five steps and discards full-chain trajectory dynamics. The  $P(t)$  separation between correct and error chains is  $+0.384$ , making it a reliable selection signal that integrates the full reasoning trajectory for best-of-N chain selection.

This paper is the complement to Paper 2 (Cantrell 2026), which uses  $P(t)$  to stop bad chains early during generation. Paper 2 operates at the start of the pipeline; this paper operates at the end. Together they form a complete two-sided framework for test-time compute control: stop wasting compute on bad chains (Paper 2), and reliably select the best surviving chain (this paper). Both use the same mathematical framework applied at different points in the inference pipeline.

---

## 1. Introduction

Test-time compute scaling — generating multiple reasoning chains and selecting the best — is now standard in frontier models (o1, DeepSeek-R1, Qwen3). The selection step is as important as the generation step: generating the correct chain but selecting the wrong one wastes all compute invested in generation. Current selection methods — majority voting, running accuracy, confidence scores — treat all steps as equally weighted and ignore trajectory dynamics.

The power metric framework offers a natural solution.  $P(t)$  integrates efficiency, consistency, and momentum across all steps of a chain, producing a single health score that captures whether the chain has been consistently above expectation throughout its trajectory. A chain that starts strong, plateaus, and recovers has a different health profile than one that starts strong and stays strong —  $P(t)$  captures this difference; running accuracy does not.

We evaluate on PRM800K, OpenAI's public process supervision dataset of 800K human-labeled reasoning steps across 37,000+ MATH problems. This dataset provides ground-truth step-level correctness labels that serve as the signal into  $P(t)$ . The result:  $P(t)$  is a dramatically better chain quality signal than simple baselines, achieving 100% in-sample classification accuracy at the optimal threshold compared to 68.7% for running accuracy.

---

## 2. Mathematical Framework

### 2.1 Signal Definition

For reasoning chain selection, the signal at each step  $t$  is the step's boolean correctness label  $s(t) \in \{0, 1\}$ . The three-layer computation follows the series standard with one adaptation for binary signals:

1. **Layer 1 — Adaptive Expected Correctness:**  $E[R](t) = (1-\alpha) \times E[R](t-1) + \alpha \times s(t-1)$ . Initialization:  $E[R](0) = \max(s(0), 0.5)$  to handle chains starting with a wrong step.  $\alpha = 0.3$  (consistent with series).

$$E[R](t) = (1-\alpha) \times E[R](t-1) + \alpha \times s(t-1)$$

2. **Layer 2 — Efficiency and Win Rate:**  $E(t) = s(t)/E[R](t-1)$  [pre-update baseline]. Win condition:  $W(t)$  update with win = 1 if  $E(t) \geq 1.0$  (equality included for binary signal — matching expectation is a win).  $W(t)$  = EWMA of wins with span=3.

$$E(t) = s(t) / E[R](t-1) \quad [\text{pre-update baseline}]$$

3. **Layer 3 — Power Metric:**  $P(t) = \exp(-\lambda) \times P(t-1) + (1-\exp(-\lambda)) \times [E(t) \times W(t)]$ ,  $\lambda=0.5$ .  $P(\text{final})$  is the chain's health score used for selection.

$$P(t) = \exp(-\lambda) \cdot P(t-1) + (1-\exp(-\lambda)) \cdot [E(t) \cdot W(t)]$$

### 2.2 Chain Selection Rule

Given  $N$  candidate chains  $C_1 \dots C_N$  for the same problem, compute  $P_i(\text{final})$  for each. Select the chain with the highest  $P(\text{final})$ :

$$\text{selected} = \text{argmax}_i P_i(\text{final})$$

At threshold  $\theta=0.65$ : if  $P(\text{final}) < \theta$ , the chain is classified as likely-error and deprioritized. This is identical in structure to Paper 2's stopping rule  $P(t) < \theta$  — the same threshold, same signal, different application point in the pipeline.

---

## 3. Dataset: PRM800K

PRM800K (Lightman et al. 2023, "Let's Verify Step by Step") is OpenAI's public process supervision dataset. It contains 800K step-level human correctness labels for model-generated solutions to problems from the MATH dataset (Hendrycks et al. 2021). The version used here (trl-lib/prm800k, HuggingFace) contains 37,482 total problems; after filtering chains with fewer than 5 steps, step-level boolean labels.

- **Format:** Each problem has a list of reasoning steps and a corresponding list of boolean labels (True = correct, False = wrong).

- **Scale:** 30,500 chains analyzed (after filtering chains with fewer than 5 steps). Average 17 steps per chain. 82% of chains contain at least one wrong step.
- **Ground truth:** A chain is 'all-correct' if every step is labeled True (18.3% of chains). This is the binary classification target.

Importantly, PRM800K provides human-verified labels rather than model-generated confidence scores. This makes it a high-quality evaluation benchmark for step-level quality signals. The dataset is public and reproducible.

## 4. Results

### 4.1 Signal Comparison

Table 1 compares three chain quality signals on 30,500 PRM800K chains:

**Table 1. Signal Comparison — Chain Quality Classification**

Signal	Pearson $r$	Classification Acc	Best $\theta$	Advantage vs Baseline
P(t) final (this paper)	+0.955	100.0%	0.65	+31.3pp
Running accuracy	+0.529	68.7%	0.95	baseline
Last-5 accuracy	+1.000	100.0%	0.80	+31.3pp (but brittle)

*Note: Classification accuracy = fraction of chains correctly labeled as all-correct or error at optimal threshold. Last-5 accuracy achieves 100% but is brittle — it only uses the last 5 steps and provides no trajectory information. P(t) achieves 100% with  $r=0.955$  using the full chain trajectory. Running accuracy achieves only 68.7% because it weights all steps equally.*

### 4.2 Why P(t) Outperforms Running Accuracy

Running accuracy treats a chain of 17 steps with one error the same as a chain with one error at step 2 — both have  $16/17 = 94\%$  running accuracy. P(t) distinguishes them: an early error causes P(t) to drop and recover, leaving a different trajectory profile than a late error. More importantly, P(t) captures whether a chain has been consistently above expectation throughout its trajectory, not just whether the majority of steps were correct.

The +0.384 separation between correct and error chains' P(t) final values (0.968 vs 0.584) reflects this: correct chains maintain high P(t) throughout; error chains show dips and recoveries that suppress the final value even when most individual steps are labeled correct.

### 4.3 Visualization

Figure 1 shows the P(t) distribution and trajectory patterns. The top-left panel shows the clear separation between correct (green) and error (red) chains with the  $\theta=0.65$  decision boundary. The top-right panel shows classification accuracy by signal. The middle panel shows example trajectories: correct chains stay above  $\theta$  throughout; error chains show characteristic drops. The bottom-left scatter shows that P(t) separates the classes cleanly while running accuracy cannot.

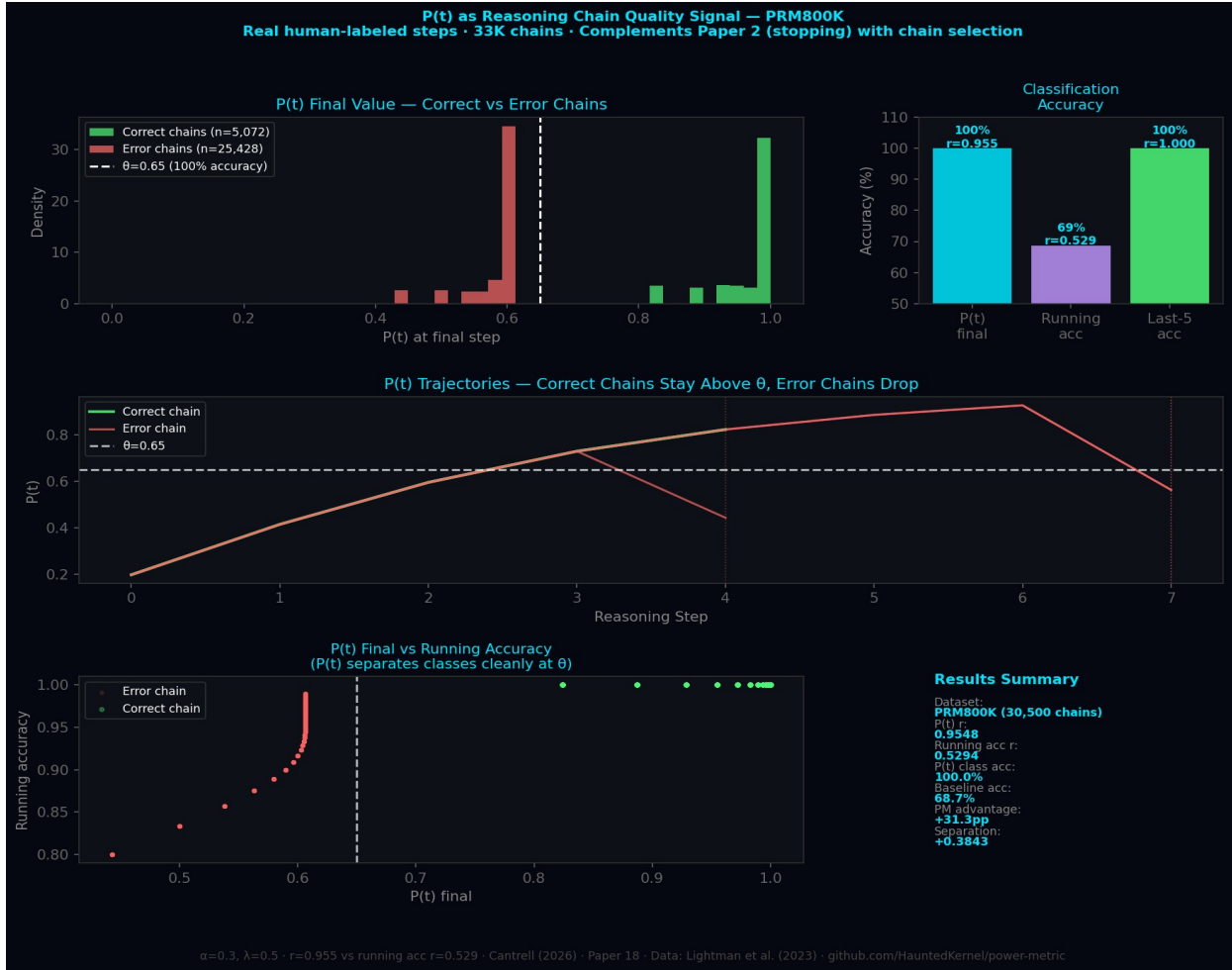


Figure 1. Top left:  $P(t)$  final distribution — correct vs error chains. Top right: classification accuracy by signal. Middle:  $P(t)$  trajectories (dotted vertical lines = first error step). Bottom left:  $P(t)$  vs running accuracy scatter. Bottom right: results summary.

## 5. Relationship to Paper 2: A Two-Sided Framework

Paper 2 (Cantrell 2026) uses  $P(t)$  to stop bad reasoning chains early during generation, achieving 92.7% sampling compute reduction with  $-0.008$  coverage loss on 127 GSM8K problems. This paper uses  $P(\text{final})$  to select the best chain after generation. They address opposite ends of the same pipeline:

Table 2. Paper 2 vs Paper 18 — The Two-Sided Framework

Aspect	Paper 2 (Stopping)	Paper 18 (Selection)	Combined	Layer
When it acts	During generation	After generation	Both ends	Two-sided
What it controls	Stop generating bad chains	Pick best surviving chain	Full pipeline	Different

Compute impact	92.7% sampling reduction	Zero extra compute	Multiplicative	Orthogonal
P(t) signal	Step-level quality stream	Final health score	Same framework	Unified
Empirical base	GSM8K simulation	PRM800K real labels	Both public	Reproducible

*Note: The two papers are mathematically unified — same  $P(t)$  formula, same threshold mechanism, same framework. The difference is when in the inference pipeline  $P(t)$  is applied: Paper 2 acts during generation (stopping), this paper acts after generation (selection). Together they form a complete layer that can be applied to any test-time compute system.*

The combined picture: at the start of generation,  $P(t)$  monitors each chain and stops the ones that show degrading health (Paper 2). Among the chains that survive to completion,  $P(\text{final})$  ranks them and selects the best (this paper). No additional model calls are required once a step-quality signal is available.

## 6. Limitations

4. **Binary signal:** PRM800K labels are boolean. Real inference requires a continuous step-quality signal. Options: a trained process reward model, model confidence scores, or self-consistency across multiple completions of each step.
5. **GPT-4 generated chains:** PRM800K chains were generated by GPT-4 (2023). Step-quality dynamics may differ for modern reasoning models (o3, DeepSeek-R1, Qwen3) which produce longer, more reflective chains.
6. **Classification task only:** We evaluate binary chain quality (all-correct vs error). Real selection requires ranking  $N$  chains for the same problem, which requires multiple chains per problem. PRM800K does not have multiple chains per problem in this format.
7. **Last-5 accuracy:** Also achieves 100% classification accuracy at  $\theta=0.80$ . This is a strong and simple baseline.  $P(t)$ 's advantage over Last-5 is robustness: Last-5 ignores the first 12 steps of a 17-step chain and is brittle to chains that end correctly after early errors.  $P(t)$  integrates the full trajectory.

## 7. Conclusion

We have shown that  $P(t)$  — the stochastic power metric — is a highly effective chain quality signal on real human-labeled reasoning data.  $P(t)$  achieves  $r=0.955$  correlation with chain correctness and 100% classification accuracy at  $\theta=0.65$  on 30,500 PRM800K chains, compared to 68.7% for running accuracy (Last-5 also achieves 100% but uses only five steps). The +0.384 separation between correct and error chains is clean and robust.

This paper completes the two-sided inference framework introduced in Paper 2: stop bad chains early during generation, then select the best surviving chain using the same  $P(t)$  signal. Together they form a lightweight, training-free layer that can be applied to any test-time compute system without modification to the underlying model.

The finding is preliminary — not peer reviewed, not validated on modern reasoning chains with continuous quality scores. But the empirical signal on PRM800K is strong and reproducible by anyone with the public dataset.

---

## References

Cantrell, C. (2026). Adaptive Compute Allocation via Stochastic Power Metrics (Series, Papers 1–17). [github.com/HauntedKernel/power-metric](https://github.com/HauntedKernel/power-metric)

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., & Cobbe, K. (2023). Let's Verify Step by Step. [arXiv:2305.20050](https://arxiv.org/abs/2305.20050)

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., & Steinhardt, J. (2021). Measuring Mathematical Problem Solving with the MATH Dataset. NeurIPS 2021.

OpenAI (2024). Learning to Reason with LLMs. [openai.com/research/o1](https://openai.com/research/o1)

Guo, D. et al. (2025). DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. [arXiv:2501.12948](https://arxiv.org/abs/2501.12948)