TEMPO: MULTI-TIMESCALE VALUE LEARNING AND POLICY CREDIT FOR LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for language models must turn response-level rewards into credit for individual tokens. Inspired by heterogeneous temporal value signals in the brain, TEMPO trains a multi-timescale critic, normalizes each horizon's advantages by their root-mean-square amplitude, and combines them for policy optimization. We analyze the reward kernel induced by this construction and distinguish temporal prediction from actor readout. In the original Qwen3-14B comparison, TEMPO improves greedy accuracy on all five mathematical benchmarks, including MATH500 from 78.4% to 84.0% and MATH5000 from 79.40% to 85.18%. On CommonsenseQA, Qwen2.5-7B-Instruct reaches $83.95 \pm 0.54\%$ validation accuracy across three training seeds, compared with $82.69 \pm 0.91\%$ for PPO, while generating substantially shorter answers. Controlled Qwen3-1.7B experiments then isolate the readout choice: with the same four-horizon target design, long-horizon readout achieves 72.2% MATH500 accuracy versus 64.8% for RMS averaging; a duplicated long-horizon control achieves 72.8%. Shared-pool sampled evaluation and logged advantage geometry clarify this separation. The results show why amplitude balance and temporal diversity must be evaluated alongside the reward weighting supplied to the actor, providing a concrete account of when a multi-timescale construction changes language-model policy learning. Code is available at https://anonymous.4open.science/r/TEMPO-5D50.

1 INTRODUCTION

Reinforcement learning for language models commonly evaluates an entire generated response and uses the resulting reward to update many token decisions. Value estimation helps construct the token-level credit signal, but the temporal assumptions in that construction deserve separate attention. In critic-based methods such as PPO (Schulman et al., 2017), a discount factor determines how rapidly a final reward attenuates as its distance from an earlier token increases. A collection of discount factors consequently supplies a collection of predictive targets, rather than a unique prescription for the policy update.

Multi-timescale reinforcement learning in the brain motivates considering heterogeneous temporal predictions (Masset et al., 2025). Translating this computational idea to language-model post-training requires specifying both what the critic predicts and what the actor reads. TEMPO (Timescale-Equalized Multi-horizon Policy Optimization) implements multiple value heads and computes a generalized advantage estimate for each horizon (Schulman et al., 2016). It then divides each advantage stream by its masked root-mean-square (RMS) amplitude. This operation controls relative numerical scale before a scalar actor advantage is formed.

The distinction between prediction and readout leads to a concrete question: when several normalized advantage streams disagree, should they contribute equally to the policy update? Equalizing amplitude does not establish that the streams encode interchangeable credit for a sparse terminal objective. Moreover, inverse-RMS weighting changes the reward kernel implicit in the combined advantage. These observations motivate analyzing the readout itself, in addition to comparing end-to-end training recipes.

The original mathematical evaluations show gains over PPO on all five benchmarks for Qwen3-14B, alongside competitive Qwen2.5-7B-Instruct results on three AIME splits. Three-seed CommonsenseQA training yields a modest accuracy increase and much shorter answers under TEMPO. On

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.

mathematical reasoning, four-head controls show that preserving long-horizon actor credit can matter more than mixing all available predictions. These results make a task-independent claim of benefit from temporal diversity too coarse; the policy-facing readout is part of the method.

We make three contributions. First, we give an explicit decomposition of RMS-fused advantages into a conditional reward kernel and a combined value baseline, clarifying the relation between temporal prediction and scalar policy credit. Second, we evaluate two targeted contrasts that keep the four-head critic architecture fixed: changing auxiliary prediction targets while retaining long-horizon actor credit, and changing actor readout while retaining the same set of critic targets. Third, we connect the resulting policy behavior to logged head amplitudes, cross-head agreement, and shared-pool sampled evaluation. The controlled results establish a substantial readout effect in the tested configuration; the CommonsenseQA study evaluates the broader training recipe across three seeds.

2 Related Work

Non-exponential discounting and multi-horizon advantage construction precede this work. Nafi et al. (2022) study hyperbolically discounted advantage estimation, while UGAE provides generalized advantage estimation for non-exponential discount functions (Kwiatkowski et al., 2023). Our emphasis is the effect of empirical RMS weights and actor readout in language-model PPO.

The closest readout comparison is Target Decoupling in Sun (2026), which retains auxiliary temporal predictions while updating the actor from a long-horizon advantage. We use long-head readout as an attribution control, rather than introducing it as a new algorithm. Our experiments examine sparse final-answer rewards, token-distance kernels, and four-head controls in language-model post-training.

Masset et al. (2025) motivate heterogeneous temporal value channels in biological reinforcement learning. We use that result as a computational design prior. The relevant connection is the separation between a collection of temporal predictions and the readout used to guide behavior. Our analysis characterizes that distinction in an artificial learning system; it does not fit neural recordings or assign a biological time constant to a token-level discount factor.

Value learning across horizons. Multi-horizon prediction has a substantial history in reinforcement learning. Fedus et al. (2019) relate mixtures of exponential returns to hyperbolic discounting and study multiple horizons as auxiliary prediction tasks. Gamma-nets condition value estimates on timescale (Sherstan et al., 2020), and efficient multi-horizon learning has also been studied for off-policy control (Ali et al., 2022). TEMPO uses a finite collection of return targets in language-model PPO and examines how those targets enter the actor update.

Credit assignment in language models. PPO (Schulman et al., 2017) uses a learned critic, whereas GRPO (Shao et al., 2024) derives a group-relative signal without a separate value model. Both ultimately update a token policy using scalar credit, but their estimators differ; GRPO does not itself imply a single critic discount factor. VinePPO studies credit estimation through additional continuation sampling (Kazemnejad et al., 2025). TEMPO changes the value targets and their readout while reusing the collected responses. This distinction motivates PPO as the primary attribution baseline and group-based methods as complementary algorithm comparisons.

3 TEMPO: Temporal Prediction and Policy Credit

Temporal predictors. Let s_t denote the prompt and generated prefix, a_t the next token, and r_t the reward assigned to that token. A critic with shared parameters and H output heads predicts $V_h(s_t)$ for discount factors γ_h. For each head we compute

$$\delta_t^{(h)} = r_t + \gamma_h V_h(s_{t+1}) - V_h(s_t), \tag{1}$$

$$A_t^{(h)} = \sum_{l \geq 0} (\gamma_h \lambda)^l \delta_{t+l}^{(h)}. \tag{2}$$

The implementation uses the finite sampled response and a zero value at its boundary. Each head is trained on its corresponding return target. In the controlled experiments, the value loss is summed over the four heads in every condition.

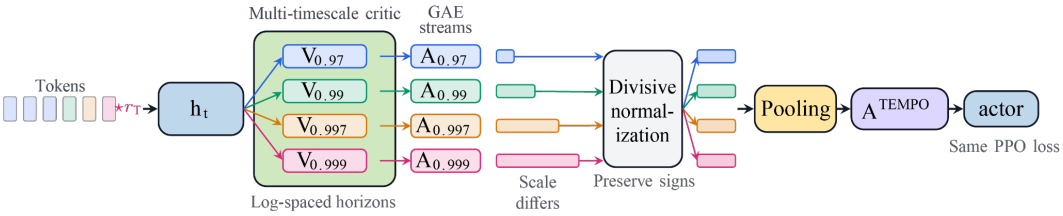

Figure 1: TEMPO's RMS-mean pipeline. A shared critic backbone supplies horizon-specific values, followed by advantage estimation, per-head RMS division, and pooling. The diagram shows the original T4 spectrum ending at $\gamma = 0.999$; the A4 controls replace its final head with $\gamma = 1$. The plotted sign preservation refers to per-head division. Rollout-wide centering and variance normalization follow pooling in the evaluated implementation.

Amplitude equalization. For the valid tokens $\mathcal{B}$ in an experience batch, define

$$\sigma_{h,\mathcal{B}} = \sqrt{\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}(A_t^{(h)})^2 + \epsilon}, \qquad U_t^{(h)} = A_t^{(h)}/\sigma_{h,\mathcal{B}}. \tag{3}$$

TEMPO's RMS-mean readout is $A_t^{\text{mean}} = H^{-1}\sum_h U_t^{(h)}$. The long-head readout is $A_t^{\text{long}} = U_t^{(H)}$, where head H has the largest discount factor. Both are followed by the implementation's common rollout-wide centering and standard-deviation normalization before the PPO update. RMS division preserves each individual head's sign before this outer normalization; the final actor signal also depends on fusion and centering.

Policy and value updates. Writing $\widetilde{A}_t$ for the final centered and normalized readout, PPO maximizes

$$L_{\text{actor}}(\theta) = \mathbb{E}_t\left[\min\left(\rho_t(\theta)\widetilde{A}_t, \text{clip}(\rho_t(\theta), 1-\eta, 1+\eta)\widetilde{A}_t\right)\right], \tag{4}$$

where $\rho_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\text{old}}(a_t \mid s_t)$ and $\eta = 0.2$. The critic targets are $G_t^{(h)} = A_t^{(h)} + V_h^{\text{old}}(s_t)$. We sum the usual clipped squared value losses over heads, using a value-clipping threshold of 0.5. The same response and task reward supply every head; the construction requires no extra rollout per horizon. It adds value outputs and return calculations while leaving token generation and the PPO clipping rule unchanged.

The reward sequence combines answer correctness with a per-token KL term relative to the reference policy. Head-specific discounts therefore act on both the terminal reward and the intermediate regularization costs. This interaction is relevant even when every head is trained on the same final answer label.

4 The Reward Kernel of RMS Fusion

With $\lambda = 1$ and zero boundary bootstrap, the finite-horizon advantage telescopes to

$$A_t^{(h)} = \sum_{u=t}^{T-1}\gamma_h^{u-t}r_u - V_h(s_t). \tag{5}$$

Conditioned on the realized RMS statistics, let $w_{h,\mathcal{B}} = 1/(H\sigma_{h,\mathcal{B}})$. Before outer normalization, the mean readout satisfies the exact identity

$$A_t^{\text{mean}} = \sum_{u=t}^{T-1} K_{\mathcal{B}}(u-t)r_u - V_{\mathcal{B}}^{\text{mix}}(s_t), \tag{6}$$

$$K_{\mathcal{B}}(d) = \sum_h w_{h,\mathcal{B}}\gamma_h^d, \qquad V_{\mathcal{B}}^{\text{mix}}(s_t) = \sum_h w_{h,\mathcal{B}}V_h(s_t). \tag{7}$$

Table 1: Mathematical benchmark results from the original evaluations. All entries are exact greedy correct counts. The upper block compares the original Qwen3-14B PPO and TEMPO checkpoints. The lower block reports Qwen2.5-7B-Instruct algorithm comparisons. AIME2024 was used for discount selection; MATH500 is a subset of MATH5000. Dashes denote entries not reported for that 7B PPO reference.

Method	AIME2024	AIME2025	AIME2026	MATH500	MATH5000
Qwen3-14B					
PPO	6/30	5/30	6/30	392/500	3970/5000
TEMPO	**7/30**	**7/30**	**8/30**	**420/500**	**4259/5000**
Qwen2.5-7B-Instruct					
PPO reference	4/30	–	–	–	–
GRPO	3/30	0/30	2/30	366/500	3583/5000
DAPO	2/30	1/30	**3/30**	362/500	3650/5000
Dr.GRPO	4/30	2/30	1/30	365/500	3583/5000
REINFORCE	3/30	1/30	**3/30**	**371/500**	**3688/5000**
TEMPO	**7/30**	**4/30**	2/30	369/500	3646/5000

For terminal task reward R and per-token KL cost $\beta\kappa_u$, the reward term becomes

$$RK_{\mathcal{B}}(T-1-t) - \beta \sum_{u=t}^{T-1} K_{\mathcal{B}}(u-t)\kappa_u. \tag{8}$$

Thus, RMS equalization changes the relative coefficients applied to terminal reward and intermediate KL costs. The normalized kernel $\bar{K}_{\mathcal{B}}(d) = K_{\mathcal{B}}(d)/K_{\mathcal{B}}(0)$ makes this relative weighting visible. Because RMS statistics depend on collected experience, this is a conditional decomposition, rather than a claim that training optimizes one fixed discount kernel.

Relation to decreasing impatience. For fixed positive weights and $0 < \gamma_h \leq 1$, the mixture satisfies $K(d+1)^2 \leq K(d)K(d+2)$. Consequently, the effective one-step discount $K(d+1)/K(d)$ is nondecreasing with delay. This property explains how exponential channels can produce fast initial attenuation followed by a slower decline. A finite mixture is not, in general, an exact hyperbolic discount function. In particular, a positive $\gamma = 1$ component gives a nonzero asymptotic plateau. This interpretation connects the biological motivation to a testable computational quantity without identifying token steps with biological time constants.

Prediction and scalar credit. Equation (6) also explains why multiple critic outputs do not eliminate scalarization: the actor still receives a scalar. If a scalar predictor exactly represented $V_{\mathcal{B}}^{\text{mix}}$ under fixed weights, it would reproduce the same algebraic advantage construction. Whether joint multi-head training learns a better predictor is a separate empirical question. The present readout controls test the behavior of multi-head training and readout choices; they do not identify the advantage of a separately trained scalar mixed-target critic.

5 Experiments

5.1 Mathematical Benchmark Results

We first report the original mathematical benchmark evaluation, then use targeted controls to study the source of the effect. Table 1 retains the complete greedy benchmark profiles for AIME2024, AIME2025, AIME2026, MATH500, and the 5,000-problem MATH test set. The models are trained on MATH and evaluated on these mathematical benchmarks. The original single-seed Qwen3-14B PPO and TEMPO runs both use 116 rollout updates and a 4,096-token training response cap; these are distinct from the later three-seed checkpoints summarized in Appendix D.

Results across mathematical benchmarks. On Qwen3-14B, TEMPO improves all five greedy metrics: it solves one, two, and two additional problems on the three AIME splits, respectively, and 28 additional MATH500 problems and 289 additional MATH5000 problems. The corresponding

Table 2: Matched four-head controls on MATH500. Qwen3-1.7B, seed 42, 59 rollout updates. A4 uses $(0.97, 0.99, 0.997, 1)$; D4 uses $(1, 1, 1, 1)$. All entries are percentages except exact greedy counts. Sampled results use a common pool of eight candidates per question within each condition.

Condition	Actor readout	Greedy	Sampled @1	@2	@4	@8
D4-long	Normalized head 4	72.8 (364/500)	73.88	79.39	83.33	86.00
A4-long	Normalized head 4	72.2 (361/500)	73.13	78.76	82.73	85.80
A4-mean	Normalized mean	64.8 (324/500)	64.28	69.51	73.55	77.20

MATH gains are 5.6 and 5.78 percentage points. On Qwen2.5-7B-Instruct, the AIME2024 PPO-to-TEMPO change is 4/30 to 7/30; the base 7B and 14B models solve 3/30 and 7/30, respectively, on that development split. Among the additional 7B algorithms, TEMPO has the highest AIME2024 and AIME2025 counts, while the MATH ordering varies across methods. These benchmark results motivate testing temporal credit construction, and the following controls distinguish critic-target diversity from the actor's readout.

5.2 Controlled Mathematical Reasoning

Design. We use three four-head conditions. D4-long predicts four copies of the undiscounted target, with $\boldsymbol{\gamma} = (1, 1, 1, 1)$, and reads the fourth normalized head. A4-long predicts $\boldsymbol{\gamma} = (0.97, 0.99, 0.997, 1)$ and reads the normalized $\gamma = 1$ head. A4-mean uses the same diverse target design and averages the four normalized advantages. The D4-long/A4-long contrast changes the auxiliary targets at fixed actor horizon. The A4-long/A4-mean contrast changes actor readout at fixed target design. Each condition is trained separately, so its trajectories and learned parameters can differ.

Training and evaluation. Our implementations build on OpenRLHF (Hu et al., 2024). All three conditions start from Qwen3-1.7B (Yang et al., 2025) with seed 42 and train for 59 rollout updates on the same 7,500-row MATH training file (Hendrycks et al., 2021). The rollout batch size is 128, with three PPO epochs, actor and critic learning rates of 10^{-6} and 9×10^{-6}, $\lambda = 1$, and initial KL coefficient 0.01. Training uses a 2,048-token response cap and a 512-token prompt cap; evaluation uses a 3,072-token response cap and a 4,096-token context limit. The learning-rate schedule is configured for two data episodes, with training stopped at the specified update count. We keep the data, prompt rendering, grader, and evaluation settings fixed across these three conditions.

Greedy accuracy and sampled pass@k are reported separately. For sampled evaluation, each question has one pool of eight responses generated at temperature 0.7 and top-p 0.95. With c_i correct responses among the eight candidates for question i, we compute

$$\widehat{\text{pass@}k} = \frac{1}{N} \sum_{i=1}^{N} \left[1 - \frac{\binom{8-c_i}{k}}{\binom{8}{k}} \right], \qquad k \in \{1, 2, 4, 8\}. \tag{9}$$

This estimator uses the same candidate pool for every k. Paired intervals resample the 500 questions 3,000 times with bootstrap seed 20260926; they characterize evaluation-question uncertainty for the fixed checkpoints. These controlled training runs use one training seed per condition. The three-seed study in Section 5.4 is reported separately. The original benchmark results appear in Section 5.1; additional normalization ablations and training configurations appear in Appendix D.

Actor readout changes the outcome. At the same diverse critic target design, A4-long reaches 72.2% greedy accuracy compared with 64.8% for A4-mean, a difference of 7.4 percentage points (paired question-bootstrap 95% interval: 4.6 to 10.6). The comparison has 48 questions solved only by A4-long and 11 solved only by A4-mean. The shared-pool estimates show the same ordering: sampled pass@1 is 73.13% versus 64.28%, and pass@8 is 85.8% versus 77.2%.

The duplicated-head control resolves attribution. D4-long reaches 72.8% greedy accuracy and 86.0% pass@8. A4-long differs from it by -0.6 percentage points in greedy accuracy, with a paired interval from -3.6 to 2.4. The controlled evidence therefore attributes the prominent separation to

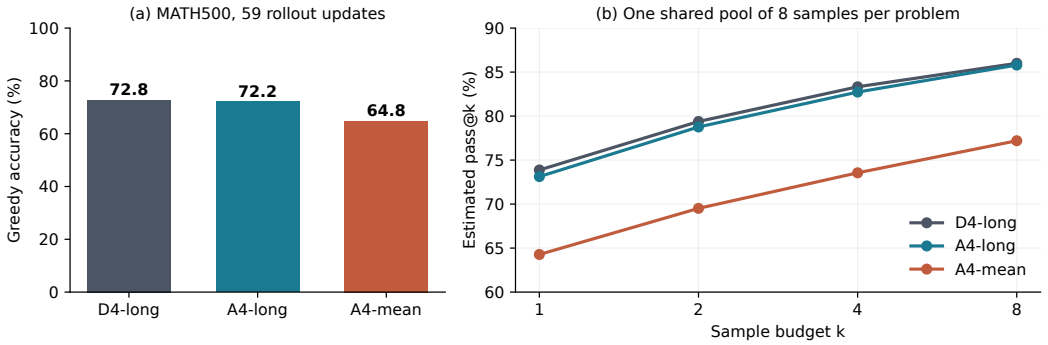

Figure 2: Actor readout controls on Qwen3-1.7B after 59 rollout updates. All conditions use four critic heads, seed 42, and the same MATH500 evaluation protocol. Greedy decoding is distinct from sampled pass@1. The sampled curves use the standard estimator from one pool of eight responses per problem.

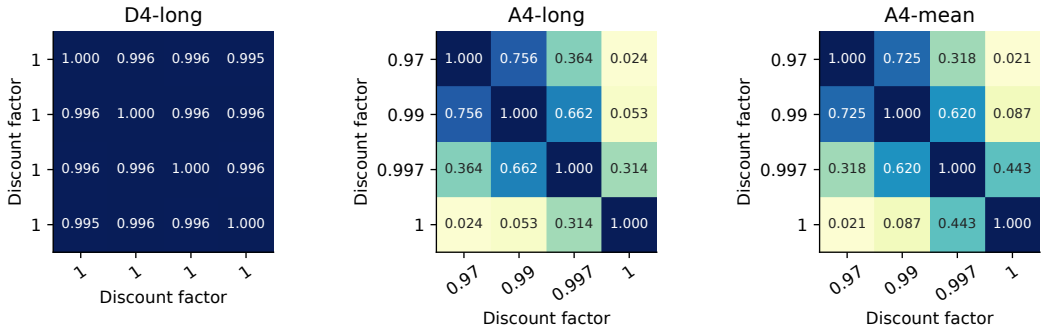

Figure 3: Mean pairwise cosine of token advantages over logged experience batches. Positive per-head RMS division does not change a batch's pairwise cosines. The panels compare temporal agreement under the three separately trained 1.7B conditions. Records are averaged equally and are not independent training seeds.

the readout contrast in this configuration, while leaving the incremental value of diverse auxiliary targets unresolved.

5.3 What Does RMS Equalization Balance?

We record diagnostics every eight rollout updates, yielding 56 experience-batch records per 1.7B condition over updates 8 through 56. The average shortest-to-longest-head advantage cosine is 0.995 for D4-long, 0.024 for A4-long, and 0.021 for A4-mean. The mean within-batch ratio of the largest to smallest head RMS is 1.01, 7.13, and 9.89, respectively. Diverse heads consequently supply substantially different advantage streams even after their amplitudes are equalized. This is evidence about agreement among token advantages, not a measurement of policy-gradient correctness.

The conditional kernels expose a second effect. Using each logged batch's inverse-RMS weights and then averaging normalized kernels, A4-mean assigns a relative terminal-reward coefficient of 0.056 at a distance of 1,024 tokens and 0.047 at 2,048 tokens. The long-head readout has a constant relative coefficient of one. The values describe the reward component of the pre-normalization advantage; value baselines, KL costs, and the subsequent PPO update also affect learning.

The completed 58-update Qwen3-14B training logs provide a second scale for these diagnostics. With $(0.97, 0.99, 0.997, 0.999)$ and RMS-mean readout, the shortest-to-longest-head cosine averages 0.122, compared with 0.993 for four repeated $\gamma = 1$ heads. These 14B results characterize the structure of training-time advantages.

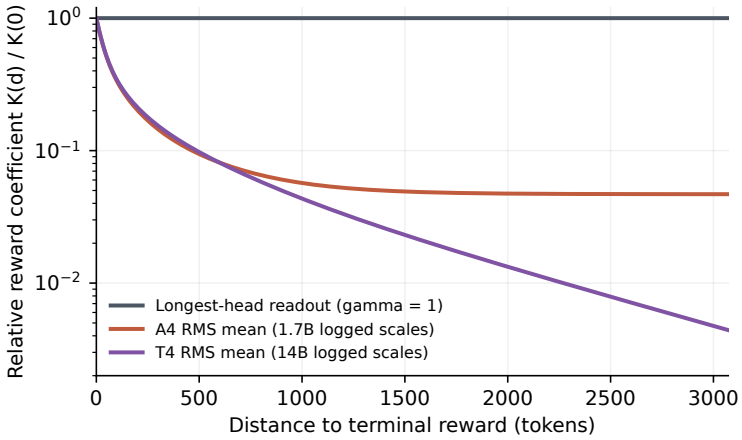

Figure 4: Analytical reward kernels conditioned on logged RMS statistics. Each batch's kernel is normalized at delay zero and the normalized curves are averaged. The horizontal curve is the relative kernel for a $\gamma = 1$ actor head. Curves describe coefficients in the advantage decomposition, not measured neural discounting or a causal performance model. The 1.7B curve is analytically extended beyond its training response cap.

Table 3: CommonsenseQA validation, Qwen2.5-7B-Instruct. Accuracy is reported as exact counts and percentages. The final row gives mean $\pm$ sample standard deviation across three training seeds. Every run is evaluated on all 1,221 questions.

	Accuracy (%)		Mean output tokens	
Seed	PPO	TEMPO	PPO	TEMPO
20	83.37 (1018)	83.87 (1024)	71.27	6.47
42	81.65 (997)	83.46 (1019)	70.55	6.09
80	83.05 (1014)	84.52 (1032)	107.84	7.82
Mean	82.69 ± 0.91	83.95 ± 0.54	83.22	6.79

5.4 Commonsense Question Answering

To examine the method beyond mathematical answers, we train Qwen2.5-7B-Instruct (Qwen et al., 2024) separately on CommonsenseQA (Talmor et al., 2019). PPO and TEMPO each use seeds 20, 42, and 80, with 75 rollout updates, a rollout batch of 128, three PPO epochs, and a 1,024-token response cap. The task reward checks the selected answer option. All six final checkpoints are evaluated greedily on the same 1,221 validation questions with the same answer extractor and generation cap. These are in-domain held-out evaluations after CommonsenseQA training, rather than zero-shot transfer from mathematical training.

TEMPO reaches 83.95% mean accuracy, compared with 82.69% for PPO, with a positive difference for each paired training seed. The mean difference is 1.26 percentage points. The TEMPO responses are also substantially shorter: 6.79 tokens on average, compared with 83.22 for PPO. Most TEMPO responses directly emit the boxed option. No response in either method reaches the generation cap, and answer parsing succeeds for all TEMPO responses and all but one PPO response across the three runs.

This result broadens the empirical setting and helps interpret temporal weighting. A multiple-choice task can be solved with a short output, so improvement need not take the form of a longer reasoning trace. Here accuracy and concise answer production occur together. We report the outcome as a comparison of the two training implementations; the four-head MATH500 experiments provide the more focused attribution of critic targets and actor readout.

6 Discussion and Conclusion

The original 14B evaluation shows improved greedy accuracy across five mathematical benchmarks, and the 7B comparisons extend the evaluation to group-based algorithms. The experiments distinguish two uses of temporal prediction. Auxiliary value targets change what the critic learns; actor readout determines which resulting advantages drive the policy. In the controlled mathematical setting, restricting readout to the longest horizon recovers 7.4 accuracy points relative to RMS averaging, without establishing an additional gain from diverse auxiliary heads. On CommonsenseQA, the complete TEMPO recipe yields a consistent positive accuracy difference across three seeds and a marked shift toward concise answers. These outcomes are compatible with a method whose effect depends on how task rewards, response length, and temporal weighting interact.

The kernel decomposition makes that dependence explicit. RMS division equalizes numerical amplitudes but also changes the coefficients of delayed rewards and KL costs. Head agreement and conditional reward kernels therefore provide useful diagnostics alongside final accuracy. The usefulness of multiple value horizons depends on both the prediction targets and how the actor uses them.

Our evidence covers fixed spectra and a small set of model and task configurations. The controlled MATH500 intervals quantify question-level uncertainty, whereas the CommonsenseQA standard deviations quantify training-seed variation. A separately trained scalar critic for the same mixture target, and replicated readout controls, would further distinguish representation learning from target weighting. The present results establish the readout distinction and provide an implementation-level account of multi-timescale credit in language-model PPO.

Reproducibility Statement

The experimental sections and Appendix B specify training budgets, discount spectra, evaluation protocols, and diagnostic definitions. The supplementary analysis files contain per-question correctness records and CPU scripts for reproducing the controlled MATH500 and CommonsenseQA tables. Proofs of the analytical identities appear in Appendix A.

AI Use Statement

The authors conceived TEMPO, designed the main experiments, analyzed the principal results, and manually prepared the original illustrations. Generative AI tools assisted with literature review, supplementary experiment design, code implementation and debugging, supplementary analyses and derivations, and manuscript drafting and editing. The authors reviewed the results and conclusions and take responsibility for the final work.

References

Raja Farrukh Ali, Nasik Muhammad Nafi, Kevin Duong, and William Hsu. Efficient Multi-Horizon Learning for Off-Policy Reinforcement Learning. In *NeurIPS Deep Reinforcement Learning Workshop*, 2022. URL https://openreview.net/forum?id=7Se_75p9dVA.

Matteo Carandini and David J. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13:51–62, 2012.

Kazuki Enomoto, Naoyuki Matsumoto, Hitoshi Inokawa, Minoru Kimura, and Hiroshi Yamada. Topographic distinction in long-term value signals between presumed dopamine neurons and presumed striatal projection neurons in behaving monkeys. *Scientific Reports*, 10:8912, 2020.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G. Bellemare, and Hugo Larochelle. Hyperbolic Discounting and Learning over Multiple Horizons, 2019. URL https://arxiv.org/abs/1902.06865.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. OpenRLHF: An Easy-to-use, Scalable and High-performance RLHF Framework, 2024. URL https://arxiv.org/abs/2405.11143.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining Credit Assignment in RL Training of LLMs, 2025. URL https://arxiv.org/abs/2410.01679.

Ariel Kwiatkowski, Vicky Kalogeiton, Julien Pettré, and Marie-Paule Cani. UGAE: A novel approach to non-exponential discounting, 2023. URL https://arxiv.org/abs/2302.05740.

Paul Masset, Pablo Tano, HyungGoo R. Kim, Athar N. Malik, Alexandre Pouget, and Naoshige Uchida. Multi-timescale reinforcement learning in the brain. *Nature*, 642:682–690, 2025. doi: 10.1038/s41586-025-08929-9.

Ali Mohebi, Wei Wei, Lilian Pelattini, Kyoungjun Kim, and Joshua D. Berke. Dopamine transients follow a striatal gradient of reward time horizons. *Nature Neuroscience*, 27:737–746, 2024.

Nasik Nafi, Raja Farrukh Ali, and William Hsu. Hyperbolically Discounted Advantage Estimation for Generalization in Reinforcement Learning. In *ICML Workshop on Decision Awareness in Reinforcement Learning*, 2022. URL https://openreview.net/forum?id=UjPGKYtzuuk.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 Technical Report, 2024. URL https://arxiv.org/abs/2412.15115.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

Craig Sherstan, Shibhansh Dohare, James MacGlashan, Johannes Günther, and Patrick M. Pilarski. Gamma-Nets: Generalizing Value Estimation over Timescale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5717–5725, 2020.

Margarida Sousa, Pawel Bujalski, Bruno F. Cruz, Kenway Louie, Daniel C. McNamee, and Joseph J. Paton. A multidimensional distributional map of future reward in dopamine neurons. *Nature*, 642: 691–699, 2025.

Jing Sun. Representation over routing: Diagnosing temporal routing pathologies in multi-timescale PPO, 2026. URL https://arxiv.org/abs/2604.13517v4. Version 4, revised May 30, 2026.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of NAACL-HLT*, pages 4149–4158, 2019. URL https://aclanthology.org/N19-1421/.

Saori C. Tanaka, Kenji Doya, Go Okada, Kazutaka Ueda, Yasumasa Okamoto, and Shigeto Yamawaki. Prediction of immediate and future rewards differentially recruits cortico-basal ganglia loops. *Nature Neuroscience*, 7:887–893, 2004.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 Technical Report, 2025. URL https://arxiv.org/abs/2505.09388.

A Reward-Kernel Identities

Finite-response decomposition. At $\lambda = 1$, expand the temporal-difference residuals for head h:

$$A_t^{(h)} = \sum_{u=t}^{T-1} \gamma_h^{u-t} \left[r_u + \gamma_h V_h(s_{u+1}) - V_h(s_u) \right] \tag{10}$$

$$= \sum_{u=t}^{T-1} \gamma_h^{u-t} r_u - V_h(s_t) + \gamma_h^{T-t} V_h(s_T). \tag{11}$$

The implementation sets the boundary value to zero. Multiplying each head by its realized inverse-RMS coefficient and summing gives the reward-kernel identity in the main text. This is an identity conditional on the collected batch, including its RMS statistics. It does not treat those statistics as action-independent constants across data collection, nor assert an unbiased policy gradient for a fixed mixture objective.

Effective discount of a positive exponential mixture. Let $K(d) = \sum_h w_h \gamma_h^d$ with $w_h > 0$ and $0 < \gamma_h \leq 1$. By Cauchy–Schwarz,

$$K(d+1)^2 = \left(\sum_h \sqrt{w_h \gamma_h^d} \sqrt{w_h \gamma_h^{d+2}} \right)^2 \tag{12}$$

$$\leq \left(\sum_h w_h \gamma_h^d \right) \left(\sum_h w_h \gamma_h^{d+2} \right) = K(d) K(d+2). \tag{13}$$

Thus $K(d+2)/K(d+1) \geq K(d+1)/K(d)$, and the corresponding effective discount rate decreases with delay. This elementary property explains the qualitative shape of a fixed positive mixture; it is not specific to TEMPO. If the largest discount is strictly below one, the tail is asymptotically exponential. If a $\gamma = 1$ component has positive weight, the normalized kernel instead approaches the total normalized weight of such components.

Why amplitude equalization does not imply agreement. Ignoring the numerical ϵ, each normalized head has unit empirical second moment on its batch. With C_{ij} denoting the cosine between valid-token advantage vectors,

$$\frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \left(\frac{1}{H} \sum_h U_t^{(h)} \right)^2 = \frac{1}{H^2} \sum_{i,j} C_{ij}. \tag{14}$$

Outer scalar normalization can restore overall magnitude but does not make distinct head vectors point in the same direction. Cosine agreement concerns advantage vectors; it is not equivalent to agreement among policy gradients, which also depend on score functions and PPO clipping.

B Evaluation and Diagnostic Details

The MATH500 files contain exactly 500 unique normalized questions in every completed condition. The paired bootstrap matches questions by a hash of normalized question text and never silently intersects different evaluation sets. The sample pool has eight Boolean grading outcomes per question, and all sampled budgets use the same pool. Greedy decoding is evaluated separately. No sorting or cumulative maximum is applied to the resulting metric values.

The diagnostic summaries give equal weight to each logged experience batch: 56 records per 1.7B condition and 112 per 14B condition, at seven logging updates. These records describe within-run behavior. In the log field named `fusion_rms`, the recorded quantity is always the RMS of the hypothetical mean of normalized heads; for a long-head actor this field is not the RMS of its actual actor readout.

Kernel curves are computed by taking inverse logged RMS values, normalizing the resulting weights within each batch, forming the exponential mixture, and then averaging curves over batches. We do not substitute the inverse of the mean RMS for the mean of the inverse-RMS weights.

Table 4: Greedy evaluation output lengths under the common 3,072-token cap. These descriptive statistics include all responses, regardless of correctness.

Condition	Mean output tokens	Length-limit finish (%)
D4-long	2047.63	30.8
A4-long	2060.94	33.0
A4-mean	2268.73	39.2

The reported accuracy is performance under this fixed generation budget. Output lengths and termination rates differ across trained policies, so the readout comparison does not isolate an effect independent of generated length.

C Training Configuration Details

Table 5: Settings for the completed controlled mathematical experiments and CommonsenseQA study. All training runs use three PPO epochs per rollout batch, actor learning rate 10^{-6}, critic learning rate 9×10^{-6}, and initial KL coefficient 0.01.

Setting	MATH controls	14B diagnostics	CommonsenseQA
Base model	Qwen3-1.7B	Qwen3-14B	Qwen2.5-7B-Instruct
Training seeds	42	42	20, 42, 80
Rollout updates	59	58	75
Rollout batch size	128	128	128
Training batch size	64	64	64
Training response cap	2048	3072	1024
Prompt cap	512	1024	512
Evaluation response cap	3072	–	1024
GAE λ	1	1	1

The mathematical controls use three H200 GPUs and a common source branch. The 14B training diagnostics use eight H100 GPUs. The A4 spectrum is $(0.97, 0.99, 0.997, 1)$; T4 is $(0.97, 0.99, 0.997, 0.999)$. D4 repeats $\gamma = 1$ in all four heads. Logarithmic spacing is a design motivation for covering a range of horizons; these finite sets are the actual values used in the experiments.

The CommonsenseQA training scripts use the baseline PPO and legacy four-head TEMPO code paths, respectively. Both configure a single training episode with at most 9,600 examples and one sampled response per prompt. PPO enables dynamic batching, while the legacy TEMPO path uses fixed microbatches. Both use the same correctness reward module, validation evaluator, greedy sampling parameters, and option extractor. These implementation differences are retained in the configuration record; the CommonsenseQA comparison is not the same controlled code-branch ablation as the mathematical readout study.

The validation prompts contain the question and answer options in the tokenizer's user-message template. The grader extracts a predicted option and compares it with the gold option. We verified that all six output files contain the same ordered list of 1,221 unique question identifiers and gold labels, and that the stored correctness flags agree with the predictions. The reported means average the three seed-level accuracies; the standard deviations use the sample convention with denominator $3 - 1$.

D Additional Mathematical Experiments

D.1 Normalization and Algorithm Comparisons

The initial Qwen2.5-7B-Instruct development experiments compared direct averaging, per-head mean/standard-deviation normalization, and RMS division under the same four discount factors. AIME2024 was used to choose the discount configuration, so its counts are development results.

Table 6 records all three AIME splits. The PPO reference solves 4/30 AIME2024 problems; the selected RMS configuration solves 7/30. These are single-checkpoint counts, distinct from the replicated CommonsenseQA experiment.

Table 6: Historical normalization comparison with four discount horizons, Qwen2.5-7B-Instruct. Entries are greedy correct counts out of 30.

Readout normalization	AIME2024	AIME2025	AIME2026
Direct average	3	2	1
Per-head mean/std	4	1	1
Per-head RMS	7	4	2

The original algorithm benchmark profiles are reported in Table 1 in the main text.

D.2 Earlier Single- and Repeated-Horizon Controls

An earlier Qwen3-1.7B experiment compared a scalar RMS-normalized critic (S1), four repeated horizons (D4), and the T4 spectrum under RMS-mean readout. Table 7 reports these results separately because the source configuration and spectrum differ from the new A4 readout controls. Increasing the update count does not reverse the ordering between D4 and T4 in these records.

Table 7: Earlier 1.7B RMS-mean configurations. Accuracy is shown as percentages and exact counts. Dashes indicate unreported evaluations.

Configuration	Updates	MATH500	MATH5000
S1	59	78.2 (391/500)	–
D4	59	77.8 (389/500)	–
T4	59	71.0 (355/500)	–
D4	116	78.4 (392/500)	79.12 (3956/5000)
T4	116	72.0 (360/500)	73.94 (3697/5000)

D.3 Historical 14B Training Recipes

The historical three-seed Qwen3-14B summaries report MATH500 accuracies of $76.93 \pm 1.62\%$ for PPO and $84.60 \pm 0.53\%$ for TEMPO, and MATH5000 accuracies of $78.55 \pm 0.75\%$ and $85.77 \pm 0.73\%$. These recipes used different training budgets: 58 updates for the PPO checkpoints and 116 for TEMPO, with response ceilings of 3,072 and 4,096 tokens, respectively. They describe the performance of those training configurations and are not used to estimate a budget-matched method effect. The original single-seed 116-update study is a separate set of checkpoints. The present paper's readout attribution rests on the matched 59-update controls in the main text.

E Biological Motivation

Temporal heterogeneity in value-related signals has been studied in humans (Tanaka et al., 2004), monkeys (Enomoto et al., 2020), and rodents (Mohebi et al., 2024; Masset et al., 2025; Sousa et al., 2025). Figure 5 summarizes the motivating idea: a population of value signals can carry information at different temporal scales. Divisive normalization is also a well-studied neural computation (Carandini and Heeger, 2012). These observations motivate the computational components of TEMPO, while the language-model experiments evaluate their algorithmic consequences.

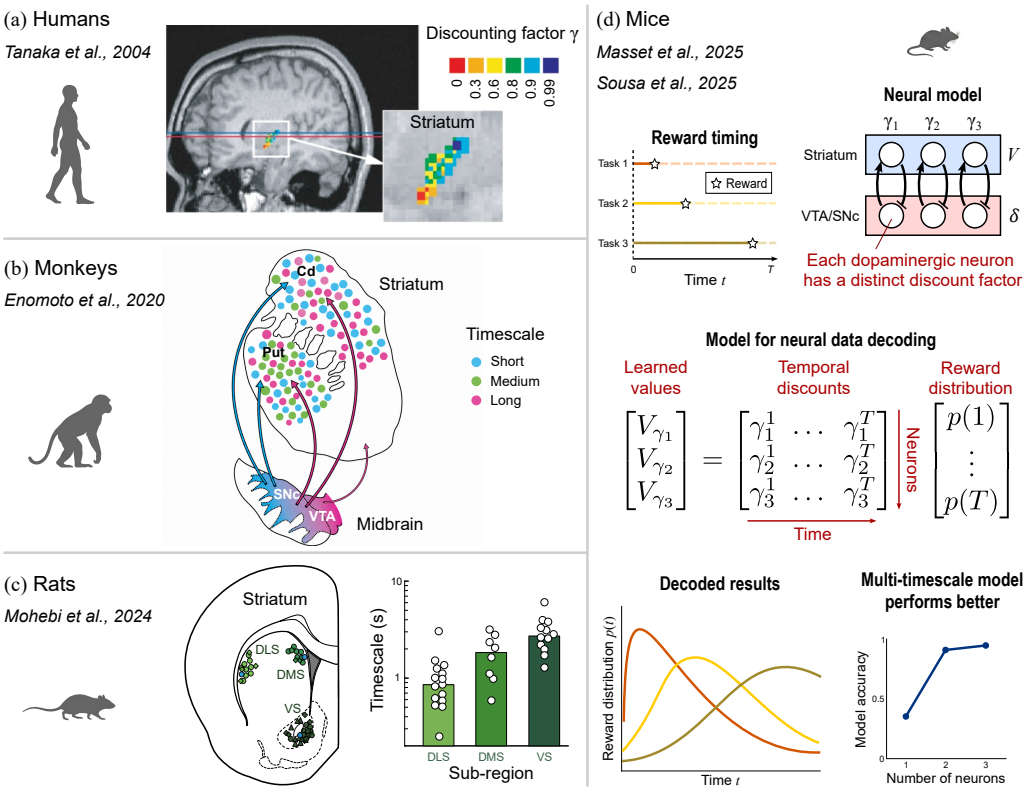

Figure 5: Overview of the biological motivation, adapted schematically from the studies named in each panel. Temporal value signals have been investigated across species and anatomical regions. The neural-model and decoding diagrams illustrate how a collection of discounted predictions can encode temporal reward structure; their schematic curves are not experimental results from TEMPO. The present work evaluates artificial token-level learning and makes no fit to these neural data.